Universidad Autónoma de Madrid
Escuela politécnica superior

Grado en Ingeniería en Tecnologías y Servicios de Telecomunicación

Trabajo Fin de Grado

BIOMETRIC AUTHENTICATION BASED ON INTERACTION WITH TOUCHSCREEN

AUTENTICACIÓN BIOMÉTRICA BASADA EN INTERACCIÓN CON PANTALLA TÁCTIL

Autor: Ada Pozo Pérez
Tutor: Julián Fiérrez Aguilar

Mayo 2017

BIOMETRIC AUTHENTICATION BASED ON INTERACTION WITH TOUCHSCREEN

AUTENTICACIÓN BIOMÉTRICA BASADA EN INTERACCIÓN CON PANTALLA TÁCTIL

Autor: Ada Pozo Pérez
Tutor: Julián Fiérrez Aguilar

Biometrics and Data Pattern Analytics Laboratory - BiDA Lab
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Mayo 2017

Abstract

Resumen

La gran popularidad de los smarthphones y el incremento en su uso para aplicaciones diariamente ha provocado que lleven información sensible, como los detalles de nuestras cuentas bancarias, contraseñas o correos electrónicos. Motivados por las limitaciones en seguridad de los sistemas tradicionales (por ejemplo, códigos PIN, patrones secretos), que pueden romperse fácilmente, se han desarrollado nuevos métodos usando biometrías para autenticar a los usuarios. Uno de estos métodos es la autenticación continua, en la cual un usuario es autenticado de forma pasiva, haciendo uso de sus biometrías. De esta manera, se garantiza la seguridad más allá del punto de acceso, asegurando que la persona que usa el dispositivo es la misma que se inscribió. Entre estos métodos para autenticación continua, este trabajo se centra en el uso de la interacción habitual de los usuarios con la pantalla táctil. Cada persona se comporta de forma diferente al deslizar los dedos por la pantalla. Teniendo en cuenta la frecuencia con la cual se efectúan las distintas operaciones, hábitos característicos, como la fuerza, el ritmo o el ángulo usados dan como resultado patrones discriminativos que se pueden usar para autenticar a los usuarios.

En el presente trabajo se exploran dos enfoques distintos para la autenticación basada en interacción con pantalla táctil: discriminativo basado en máquinas vector-soporte, y estadístico basado en mezclas de Gaussianas. Adicionalmente, se estudia un sistema basado en la fusión de los dos anteriores. La base de datos usada para el análisis se compone de datos táctiles de las operaciones más comunes, como por ejemplo los trazos hechos al deslizar un dedo por la pantalla, obtenidos de 190 sujetos. Se utiliza como referencia un artículo de la literatura, mejorando sus resultados. Usando bloques de diez trazos para la autenticación, se obtienen tasas de Equal Error Rate entre el 8% y el 22% para diferentes operaciones táctiles. Aunque el enfoque estadístico obtiene resultados ligeramente peores que las máquinas vector-soporte, es capaz de autenticar usuarios que tienen mal rendimiento en el otro sistema debido a la gran variabilidad intra-usuario. De esta forma, al fusionar los sistemas éstos se complementan entre sí. El rendimiento en distintas operaciones muestra que algunos gestos contienen más información del usuario y son más discriminativos que otros (en particular, los trazos horizontales son más discriminativos que los verticales). Los resultados experimentales muestran que las biometrías táctiles son lo suficientemente discriminativas para reconocimiento de personas y que son un método prometedor para la autenciación activa.

Palabras Clave

Autenciación activa, biometría, interacción persona-ordenador, reconocimiento de patrones, smartphone, pantalla táctil

Abstract

The great popularity of smartphones and the increase in their use in everyday applications has led to sensitive information being carried in them, such as our bank account details, passwords or emails. Motivated by the limited security of traditional systems (i.e. PIN codes, secret patterns), which can be easily broken, new methods using biometrics to authenticate users have been developed. One of these methods is active authentication, where the user is passively being authenticated in the background, based in his biometrics. This way, security in guaranteed beyond the entry point, ensuring that the person who uses the device is the same user who enrolled. Among the methods for active authentication, this work studies the users' normal interaction with touchscreens. Every person behaves differently when swiping their fingers on a touchscreen. Given the frequency in which touch operations are performed, characteristic habits, like the strength, rhythm or angle used result in discriminative patterns that can be used to authenticate users.

In the present work, we explore two recognition approaches for authentication based on touchscreen interaction: discriminative based on Support Vector Machines, and statistical based on adapted Gaussian Mixture Models. Additionally, a system based on the fusion of the two previous systems is studied. The database used for the analysis consists of touch data from the most common operations, i.e., swipes made with one finger on the screen, collected from 190 subjects. An article in the literature is used as a reference, improving its results. Using blocks of ten strokes for authentication, Equal Error Rates between 8% and 22% are obtained for different kind of touch operations. While the statistical approach obtains slightly worse performance than Support Vector Machines, it is capable of authenticating users who obtain bad performances with the other system because of large intra-user variability. That way, both systems complement each other when fusing them. The performance across different kinds of touch operations shows that some gestures hold more user-specific information and are more discriminative than others (in particular, horizontal swipes appear to be more discriminative than vertical ones). The experimental results show that touch biometrics have enough discriminability for person recognition and that they are a promising method for active authentication.

Key words

Active authentication, biometrics, human computer interaction, pattern recognition, smartphone, touchscreen

Agradecimientos

En primer lugar, quiero dar las gracias a Julián Fiérrez y a Aythami Morales por la oportunidad que me han ofrecido y por toda la ayuda que me han brindado. También quiero agradecer todo el tiempo y esfuerzo que han invertido durante este año.

También quiero agradecérselo a mi familia por todo el apoyo que me han dado. A mi hermana Bea, por su ayuda y por ofrecerse a revisar este trabajo aunque no entendiera una palabra. A mis padres, que me hicieron cuestionarme si las máquinas podían aprender y ver las cosas desde otra perspectiva.

Contents

List of figures

List of tables

Chapter 1

Introduction

This chapter includes a brief introduction to user authentication on mobile devices, followed by a review of previous works in touch biometrics and their motivation and challenges. Lastly, the organization of this essay will be summarized.

1.1 User authentication on mobile devices

Traditionally, the methods used for authentication on mobile devices have been using a password, a PIN code or a secret pattern. However, it has been proven that these methods have different problems [2, 3, 4, 5, 6], of which inconvenience is one of the most remarkables. Because of the need to authenticate each time the device is used, users tend to not use authentication in their phones at all, or to use short and weak passwords and PIN codes, because they are easier to remember and can be entered faster. Additionally, smudge attacks are capable of following the residues left on the device's screen when entering the same pattern repeatedly, thus gaining access to the device and proving that secret patterns are not secure enough. These kind of authentication methods are known as entry-point methods, in which you only authenticate once, when unlocking the device, and this authentication is not performed again until the device is once more locked. Therefore, one cannot detect intruders if the screen is left unlocked or cannot know if the person who is using the phone is the same user who authenticated in the first place.

To overcome these problems new methods using biometrics known as continuous or active authentication methods have been developed. In these systems users are authenticated periodically in the background by passively analysing their biometrics. The usage patterns are studied and compared to those of the legitimate user, blocking the device if there are not enough coincidences [4, 7]. Both physiological and behavioural biometrics have been studied for continuous authentication using the different sensors available in the device, such as capturing the user's face with the frontal camera, studying the typing dynamics, the user's gait data from the gyroscope, etc. An overview of these methods can be found in [5]. They are able to authenticate a user in a non-intrusive way, executing in real time in the background. They could either complement the traditional entry point methods or substitute them completely, removing the inconvenience of using a password or a PIN code [8].

Nevertheless, continuous authentication also has its drawbacks [5], namely the high computational cost, being unable to replace biometrics or the compromise between security, reducing the number of users considered legitimate, and frustrating legitimate users by rejecting them.

One method for continuous authentication whose results are only preliminary are touch biometrics on mobile devices. They are based on the idea that every person behaves differently from others when swiping their fingers on a touchscreen. Given the frequency and simplicity

of touch operations, characteristic habits, such as the rhythm, pressure, angle or strength used, are developed, which results in discriminative patterns that can be used for authentication.

1.2 Related works in touch biometrics

Touch biometrics use recorded touch data from swipes made by the user during basic operations on a device's touchscreen to authenticate him. Hence, they can be considered behavioural biometrics. As was proven in previous works [3, 4, 6], they present high inter-class variability, that is, touch data from different users show great differences and thus can be discriminative between them. However, they also present high intra-class variability, which means that they are not stable biometrics, and consequently, may change depending on the user's emotional state or with time, resulting in different patterns of use. Therefore, modelling a user can be difficult.

Nevertheless, an advantage of touch biometrics for authentication is that they don't require any additional sensors, as other biometrics like face recognition or gait do, because the device is already capturing the strokes on the screen. That way, data can be extracted passively from users' normal interaction with the screen.

Touch biometrics studies have mainly shown two approaches so far. The first one are authentication methods using touch gestures at an entry point. The second one is continuous authentication while the user performs different tasks on the device [2, 4, 7, 6]. In entry point systems, users' touch behaviour is only analysed using a set of predefined gestures, for example, the secret pattern on the unlock screen. Therefore, assumptions as of which finger is used or comparisons regarding the geometry of the hand, which cannot be used in continuous authentication, can be made. On the other hand, continuous authentication, in which the user can freely interact with the device, is the approach that will be considered in this work. Studies comparing image-based features have also been made, as well as fusing touch biometrics with phone movement and typing patterns and other touch information [3, 9]. Further information in touch gestures besides identity such as experience, gender and age, has been studied in [10].

Several studies have investigated whether touch data is discriminative and stable enough for authenticating users. In [4], we can see one of the earliest yet more comprehensive works on continuous authentication using touch data. A 27-dimensional feature vector was proposed, separating each swipe by phone orientation and left, right, up or down direction of every stroke. Using a database consisting of 41 users, Support Vector Machines with an RBF-kernel (SVM) and k-Nearest Neighbors (kNNs) classifiers were used for classification, resulting in a performance with Equal Error Rate (EER) between 2% and 4% for eleven consecutive strokes and 13% for one single stroke.

Most of the studies focus on single touch operations [2, 4, 6]. However, in [3] data recorded from keystroke, slide, pinch and handwriting operations is used. Using SVM with a RBF-kernel, EER of less than 10% are obtained. Additionally, they performed tests regarding touch biometrics distinctiveness and permanence, concluding that touch biometrics are distinctive between users, but not stable over time. Nevertheless, a solution to the lack of stability in touch biometrics was proposed. This solution is based on using an adaptive approach that uses the previous days to recalculate users' patterns.

In [9], a fusion of biometrics (typing, swiping and phone movement) is also studied. Extracting different feature vectors from each of these biometrics, k-NN with euclidean distance and random forest are used to classify. Accuracies up to 90% are obtained both combining the biometrics and evaluating each biometric separately and then averaging the result.

In [6], the use of SVMs, kNNs, neural networks and random forest classifiers is studied for different applications (i.e. document reading or picture viewing), as well as with free tasks.

Strokes were divided by phone orientation and left, right, up or down direction, as in [4], and different feature vectors were used in each of them. Several experiments are conducted in this article, obtaining an EER of 25% when using only one stroke and less than 5% when using more than ten strokes, concluding that the results become better when the operation length increases. An evaluation concerning the use of specific applications to authenticate is also performed, reaching the conclusion that specific tasks result in better performance than free task, with a EER of around 5% with free task and 1% with specific ones using the same methodology.

Despite the fact that these works have proven the applicability of touch data for authentication, the use of different datasets, algorithms, features, methodology, experimental conditions, etc. complicate possible comparisons between systems and an evaluation of their performance [2]. Because of this, in [2], they collected a public database consisting of touch data from 190 users during swipe operations and studied ten different algorithms to conclude which are better suited for continuous authentication of touch operations. The best three algorithms among these were SVM with RBF-kernel, random forest and logistic regression, all reporting 10% to 20% EER. They also proved that blocks of strokes gave better performance than authentication with a single stroke, result which was also obtained in other articles such as [4, 6, 10].

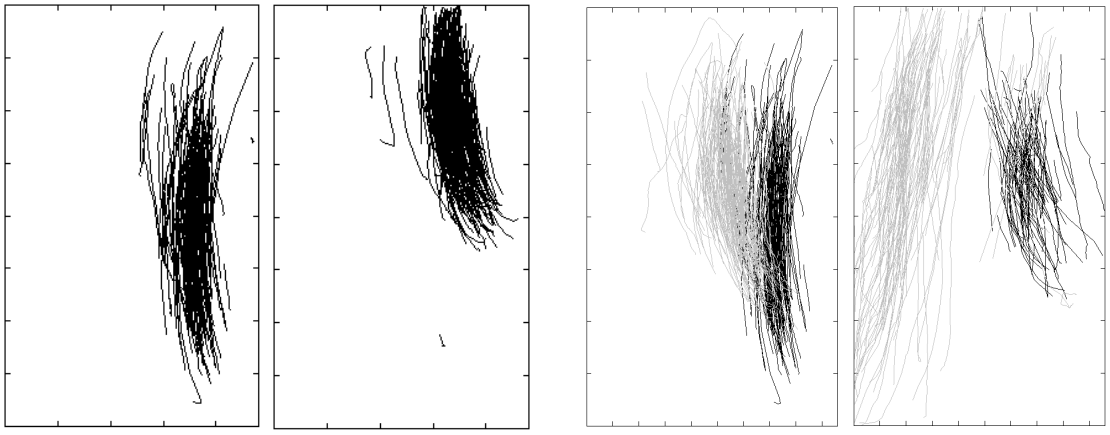
1.3 Motivation and challenges

The popularity of mobile devices, both smartphones and tablets, has grown significantly over the last few years. This growth has been accompanied by an increase in their use in everyday applications. Nowadays, our mobile devices carry our bank account details, passwords, emails, etc. Furthermore, they are more easily lost and stolen because of their portability, resulting in information leaks. Because of this, it is important to have good protections mechanisms in them. However, traditional entry point systems are not safe enough, as passwords and PIN codes tend to be shorter and easier to remember, but also easier to break, and secret patterns are vulnerable to smudge attacks. In addition, in our society, there is an increasing requirement to reliably authenticate individuals in many applications as for example, financial transactions. As a result, there is a growing body of literature looking for new ways of authentication based on the user's biometrics [5, 8]. Among these methods is continuous authentication, which periodically authenticates the user and thus ensures security in the device beyond the entry point.

One of the most active fields of investigation in continuous authentication are touch biometrics. Touch biometrics allow a passive authentication of the user that does not need any extra sensors in the device and that can obtain the data from the user's usual interaction with the touchscreen, without needing any specific task to be done. Using the swipes made in the screen, features vectors are obtained and compared to the user's template. If the score is below a user-defined threshold, the device is blocked.

One of the challenges of touch biometrics is that even though it has high inter-class variability, that allows user discrimination, it also presents high intra-user variability. This means that users' biometrics are not stable and can change over time, resulting in different patterns and making it more difficult to model users' behaviour. An example of the strokes captured for two different users, showing that they can be discriminative, can be found in figure 1.1a, while figure 1.1b is an example of the lack of stability of touch biometrics, showing how the touch data has changed in different days, for two given subjects.

Even though these problems have been addressed in the works presented in section 1.2, the results are yet preliminary. In addition, the lack of public databases combined with the different conditions and performance metrics used in each experiment complicates comparisons between different works and conclusions of which algorithms and feature vectors perform better.



(a) Example of inter-user variability, showing the strokes captured for two different users

(b) Example of intra-user variability showing data captured for two users in different sessions

Figure 1.1: Multiple strokes of different users and within the same user

1.4 Objectives

This work is focused in the study of different authentication techniques based on gestures made when interacting with a touchscreen. The objectives are:

- Implementing methods of classification for authenticating users using the public database from [2].
- Analysing the principal characteristics of strokes from interaction with a touchscreen, extracting different feature vectors based on the literature, both from touch biometrics literature [2] and from signature biometrics literature [1]. In addition, dimensionality reduction will be performed to remove information that is not discriminative between users.
- Obtaining a benchmark for comparison following the procedure described in [2] using Support Vector Machines.
- Modelling the behaviour of users using a statistical approach, adapting a universal model that represents all users, obtained with Gaussian Mixture Models, to each user's data.
- Comparing the limitations present in both systems and developing another system complementing the information found in each of them.

1.5 Organization

In chapter 2, a brief theoretical summary of the methods and classifiers used is presented. Chapter 3 contains a description of the database and a characterization of the feature vectors extracted for the experiments. The experimental protocol and results are given in chapter 4. In chapter 5 the conclusions drawn and future work are described.

Chapter 2

Methods

This chapter presents an example of the typical architecture found in a biometric system that uses gestures on a touchscreen to authenticate users. A brief summary of the classifiers and feature selection techniques used will also be presented.

2.1 Architecture of an authentication system

Authentication systems in mobile devices share a common architecture, which can be found in figure 2.1. Usually, they follow the next steps:

1. Data acquisition: touch data is obtained by recording strokes during the usual interaction with the touchscreen. For each touch point typically the x and y coordinates, the pressure, the timestamp, the area occluded by the finger and the device orientation are recorded.

2. Pre-processing and data cleansing: sequences of points are divided in strokes and short strokes that correspond to tapping motions are removed.

3. Feature extraction: each stroke is described by a feature vector that holds its most relevant and discriminative characteristics, for example its velocity, angle or length.

4. Score computation and decision: scores are obtained by comparing the feature vector extracted in the previous step with the claimed user model. Several feature vectors from different strokes, averaging their results, or only one feature vector from one stroke may be used. If the score is above a user-defined threshold the device continues processing new data, but if it's below, the device is blocked.

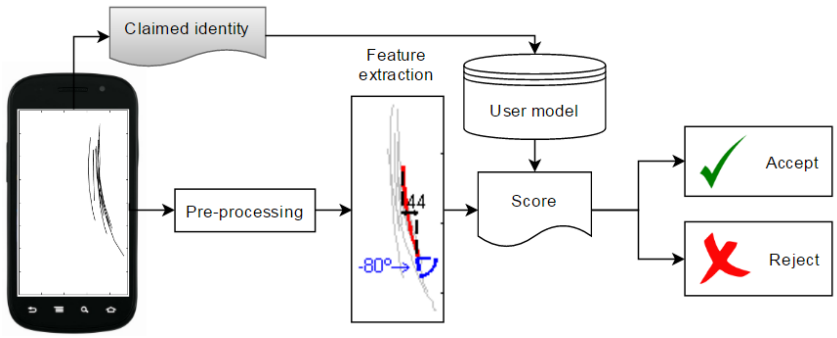


Figure 2.1: Typical system for authentication in touch biometrics

2.2 Classifiers

In this work two classifiers will be used. The first one is discriminative, based on Support Vector Machines (SVM) with a RBF-kernel, widely used in the literature [2, 3, 4, 6, 10]. The second one is statistical, based on Gaussian Mixture Model (GMM) with Universal Background Model (UBM) adaptation, used among other problems in speaker and signature verification [11].

2.2.1 Support Vector Machines

Support Vector Machines (SVM) try to find the hyperplane that best separates two classes by leaving the maximum margin from both of them. The margin is defined as the distance from the nearest point of each class to the hyperplane. In other words, the margin is the minimum distance between the hyperplane and each of the classes.

The hyperplane in the l-dimensional feature space is defined as follows [12]:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$$

where $\boldsymbol{w} = [w_1, w_2, ..., w_l]^T$, known as the weight vector, is the direction and w_0, known as the threshold, is the exact position of the hyperplane. If the hyperplane is scaled so one of the classes is indicated by $g(\boldsymbol{x}) \geq 1$ and the other one by $g(\boldsymbol{x}) \leq -1$, the margin is

$$\frac{|1|}{||\boldsymbol{w}||} + \frac{|-1|}{||\boldsymbol{w}||} = \frac{2}{||\boldsymbol{w}||}$$

which results in the following cost function:

$$J(\boldsymbol{w}, w_0, \xi) = \frac{||\boldsymbol{w}||^2}{2} + C\sum_{i=1}^{N_s}\xi_i \tag{2.1}$$

where C is known as the regularization parameter to avoid overfitting. C is a trade-off between maximizing the margin and minimizing the outliers, because of the non-separability of the classes. N_s is the number of support vectors and ξ_i are the slack parameters, which are $\xi_i > 0$ if the data point $\boldsymbol{x}_i$ is not separable between classes.

The objective, then, is to compute the parameters $\boldsymbol{w}, w_0$ so the cost function in equation 2.1 is minimized. This can be done using the Karush-Kuhn-Tucker conditions, obtaining the following result:

$$g(\boldsymbol{x}) = \sum_{i=1}^{N_s}\lambda_i, y_i{\boldsymbol{x}_i}^T\boldsymbol{x}_i + w_0 + +C\sum_{i=1}^{N_s}\xi_i$$

Classification is done depending if the previous equation is positive or negative, in which y_i is 1 if data point $\boldsymbol{x}_i$ belongs to the first class, or -1 if it belongs to the second one.

To develop complex non-linear classifiers SVM are adapted using kernels. Kernels map the input l-dimensional feature space into a k-dimensional space.

The kernel used for mapping in this work will be a Radial Basis Function (RBF):

$$K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{z}||^2}{\sigma^2}\right)$$

and the resulting classifier will be:

$$g(\boldsymbol{x}) = \sum_{i=1}^{N_s} \lambda_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + w_0 + C \sum_{i=1}^{N_s} \xi_i$$

$$\text{assign } x \text{ in } \begin{cases} 1, & \text{if } g(\boldsymbol{x}) > 0 \\ 0, & \text{if } g(\boldsymbol{x}) < 0 \end{cases}$$

Thus, there will be two parameters to adjust: C, the regularization parameter, and σ^2, the Kernel's variance. C controls how much regularization is used. If C is big, the hypothesis will have higher bias and will be more prone to overfitting, whereas a small C will mean higher bias and being more prone to underfitting. σ^2 is the width of the Gaussian kernel, so if it is large, $K(\boldsymbol{x}, \boldsymbol{z})$ will vary slower and the hypothesis will have higher bias and lower variance, tending to underfit data. On the other hand, if σ^2 is small, $K(\boldsymbol{x}, \boldsymbol{z})$ will vary more abruptly and have lower bias but higher variance, being more prone to overfitting.

2.2.2 Gaussian Mixture Models and Universal Background Model

An authentication problem with touch biometrics can be represented as a two-class classification problem with two hypothesis:

H_C: the strokes have been made by claimed user C
$H_{\overline{C}}$: the strokes have not been made by the claimed user C

Therefore, the decision is taken with a log-likelihood ratio test between hypothesis using θ as a fixed decision threshold and $\boldsymbol{x}$ as the input feature vector [11]:

$$\log p(x|H_C) - \log p(x|H_{\overline{C}}) \begin{cases} > \log\theta, & \text{accept } H_C \\ < \log\theta, & \text{reject } H_C \end{cases}$$

Usually, the hypothesis H_C is represented by a model λ_C that characterizes the user C in the feature space of $\boldsymbol{x}$, while $H_{\overline{C}}$ is represented by $\lambda_{\overline{C}}$, that models the rest of possible users. Thus, a normalized score can be obtained with the likelihood function for each of the models:

$$s = \log p(\boldsymbol{x}|\lambda_C) - \log p(\boldsymbol{x}|\lambda_{\overline{C}})$$

Gaussian Mixture Models

A Gaussian Mixture Model (GMM) models the statistic distribution λ_C as a lineal combination of d-dimensional Gaussian probability density functions [11]:

$$p(\boldsymbol{x}, \lambda_C) = \sum_{i=1}^{N} w_i p_i(\boldsymbol{x})$$

where

$$p_i(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_i|^{1/2}} \exp\left\{ -\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i) \right\}$$

The parameters of the density model are $\{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$: w_i are the mixture weights, which must satisfy $\sum_{i=1}^{N} w_i = 1$, $\boldsymbol{\mu}_i$ is a mean $d \times 1$ vector and $\boldsymbol{\Sigma}_i$ is a $d \times d$ covariance matrix, where $i = 1, ..., N$ and N is the number of Gaussian components specified for the system.

In other words, N components contribute to the formation of the distribution $p(\boldsymbol{x}, \lambda_C)$, where any of the points $\boldsymbol{x}$ can be drawn from any of the components. If the model parameters are appropriate and the number of components is high enough it can be proven that a GMM can approximate any continuous density function [12].

The model parameters are trained, given a set of training vectors, using the iterative expectation-maximization (EM) algorithm [12], which consists of two steps. In the expectation step (E-step), a function for the expectation of the log-likelihood is evaluated using the current parameters. In the maximization step (M-step), the parameters maximizing the expected log-likelihood found in the E-step are computed. Thus, the EM algorithm refines the GMM parameters by increasing in each iteration the likelihood of the estimated model for the feature vectors.

Universal Background Model

An approximation to the distribution $p(\boldsymbol{x}|\lambda_{\overline{C}})$ is obtained using a Universal Background Model (UBM). The UBM is a GMM that represents a general model. It is trained with a representative pool of feature vectors from different users.

User's model λ_C can be obtained from the UBM by combining the UBM and the user's feature vectors. The idea behind this is to adapt the UBM, deriving the user's model. To do so, the UBM parameters are updated using modified MAP (*Maximum A Posteriori*) estimation.

The following steps are used for the adaptation of the UBM [11]:

1. Given the UBM, probabilistic alignment of the user training vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_T$ into the UBM is computed:
$$P(i|\boldsymbol{x}_t) = \frac{w_i p_i(\boldsymbol{x}_t)}{\sum_{j=1}^{N} w_j p_j(\boldsymbol{x}_t)}$$
which is the posterior probability of each of the N components in the UBM for the training vector $\boldsymbol{x}_t$. Then, the sufficient statistics for weight, mean and variance are computed:
$$n_i = \sum_{t=1}^{T} P(i|\boldsymbol{x}_t)$$
$$E_i(\boldsymbol{x}) = \frac{1}{n_i} \sum_{t=1}^{T} P(i|\boldsymbol{x}_t)\boldsymbol{x}_t$$
$$E_i(\boldsymbol{x}^2) = \frac{1}{n_i} \sum_{t=1}^{T} P(i|\boldsymbol{x}_t){\boldsymbol{x}_t}^2$$
where $\boldsymbol{x}^2 = \boldsymbol{x}\boldsymbol{x}^T$ and $\boldsymbol{x}^T$ indicates transpose.

2. The adapted parameters of mixture i are computed with the new sufficient statistics:
$$\hat{w}_i = [\alpha n_i/T + (1-\alpha)w_i]\gamma$$
$$\hat{\boldsymbol{\mu}}_i = \alpha E_i(\boldsymbol{x}) + (1-\alpha)\boldsymbol{\mu}_i$$
$$\hat{\boldsymbol{\Sigma}}_i = \alpha E_i(\boldsymbol{x}^2) + (1-\alpha)(\boldsymbol{\Sigma}_i + {\boldsymbol{\mu}_i}^2) - {\hat{\boldsymbol{\mu}}_i}^2$$
where $\boldsymbol{\mu}^2 = \boldsymbol{\mu}\boldsymbol{\mu}^T$. γ is a normalization parameter that ensures that $\sum_{i=1}^{N} \hat{w}_i = 1$. The contribution of the old and new parameters is controlled by the adaptation coefficient α:
$$\alpha = \frac{n_i}{n_i + r}$$

r is known as the relevance factor. When a mixture component has low n_i, that is $n_i << r$, $\alpha \to 0$ and the GMM will barely be affected by the user's parameters. However, if n_i is high, that is $n_i >> r$, $\alpha \to 1$ and the parameters will be considerably adapted to the user's feature vectors. Thus, the relevance parameter r controls how much the mixture is adapted to new data. If r is high the user's GMM will be more similar to the UBM, while if it's low it will adapt to the user feature vectors more.

The advantage of using UBM adaptation for calculating the user model is that it 'couples' the user's model and the UBM, so a comparison between the usual behaviour and the user characteristic habits can be made.

2.3 Feature selection

When using statistical classifiers one of the main problems that degrades their performance is the high dimensionality of the feature vector. A higher number of dimensions causes a computational complexity increase, which may not result in a better performance. Even though two features carry discriminative information for classification, if they are highly correlated, using them together does not improve results [12]. Another reason are the generalization properties of the classifier. If the number of dimensions is very large, the classifier tends to not generalize well, which is a problem in touch biometrics due to the high variability of users' strokes over time.

Feature selection tries to reduce dimensionality by choosing the features that take distant values in different users but close values in the same user. In other words, it selects features with large between-class distance and small within-class distance.

In this work, Sequential Forward Floating Search (SFFS) will be used for feature selection.

2.3.1 Sequential Forward Floating Search

This algorithm follows the next steps to find the best subset of N features [13]. Given a set of F features, the objective is to find the best performing subset of N features according to a criterion C, where $N \leq F$. Let $X_n = \{x_1, x_2, ..., x_n\}$ be the best possible combination of n features and Y_{F-n} the remaining set of $F - n$ features. Knowing that the best sets of lower dimensions $X_1, X_2, ..., X_{n-1}$ are saved, three steps are performed until the N best features have been selected:

1. Inclusion: According to the optimization criterion C, the element x_{n+1} from Y_{F-n} that obtains the best performance when combined with those of X_n is selected. Then, it is added to the best subset of $n + 1$ features, $X_{n+1} = \{X_n, x_{n+1}\}$.
2. Test: First, the feature x_r that has the least cost on the criterion C when removed from X_{n+1} is chosen. If x_r is the one added in step 1, remove it and go back to that step. If it is not and the removal of another feature does not improve the criterion on the previous X_n set, go back to step 1.
3. Exclusion: Remove from X_{n+1} feature x_r to get X'_n. Following the same procedure as in step 2, find the feature x_s that has the least cost when removed from X'_n. If removing this feature does not improve the criterion that was found on the previous X_n set, go back to step 1. If it does improve, remove the feature and start over step 3.

This algorithm does not guarantee obtaining the best subset, but it usually results in an improved performance [12].

Chapter 3

System configuration

This chapter includes the description of the database used in this work as well as the systems used. Two different feature vectors are extracted: one previously used for authentication with touch interaction in [2], and another adapting the 100-dimensional feature vector presented in [1] for online signature verification.

3.1 Database

One of the most important things in any biometric recognition problem is the acquisition of data for analysis. Thus, publicly available databases are essential and allow performance comparison of different algorithms. In this work the dataset obtained in [2] will be used.

According to [2], this database consists of data from 190 users, all of whom are students, faculty or staff at Louisiana Tech University. The data was collected over two sessions, at least one day apart, using for all users only one phone model (*Google Nexus S*), running on Android 4.0. Two Android applications were used for data collection, in which users answered a series of multiple choice questions. These questions where different in each session and allowed the user to move freely, scrolling through the short paragraphs and/or images, on which questions were based. Both portrait and landscape orientation of the phone were allowed.

For each point of a stroke made by a user the application recorded the x and y coordinates, the pressure on the screen, the area occupied by the finger on the screen, the timestamp of the data point and the phone orientation. Only two types of interactions were recorded: horizontal strokes and vertical strokes. All other gestures, such as zooming or rotations, were ignored.

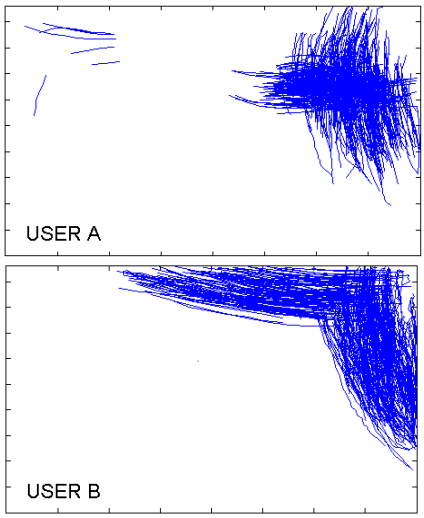
An example of the strokes captured can be found in figure 3.1.

3.1.1 Dataset analysis

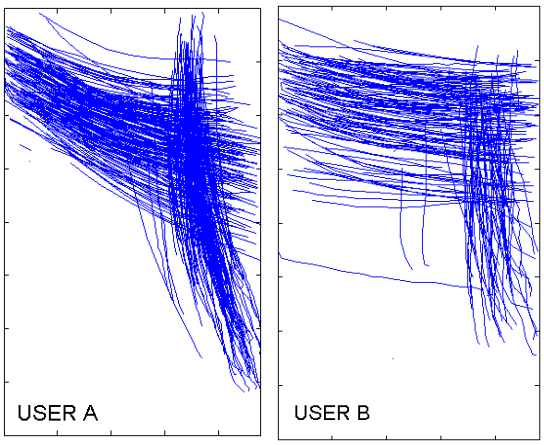
For each of the different gestures the number of users with data, the mean number of strokes and the mean number of points per stroke is summarized in table 3.1. It can be observed that most users have data for portrait orientation, while only 54 different users out of the 190 subjects have strokes from landscape orientation. The low number of users with horizontal data in landscape orientation originates that results in that condition are less reliable, because a lower number of comparisons is made. The mean number of strokes per user is around 80, except in the downwards gestures, which are the most frequently made. It is worth noting that, in both orientations, the number of points per stroke is only of about 8 points for horizontal strokes and 20 for vertical ones. This means that the active area available does not affect the number of

Table 3.1: Summary of the number of users, test samples per user and data points per stroke

			Number of users	Mean number of strokes per user	Mean number of points per stroke
Portrait	**Vertical**	**Upwards**	124	85.2	21.07
		Downwards	132	116.83	24.82
	Horizontal	**Leftwards**	104	70.23	9.75
		Rightwards	118	86.48	8.91
Landscape	**Vertical**	**Upwards**	54	80.91	18.37
		Downwards	54	122.94	21.84
	Horizontal	**Leftwards**	17	70.59	7.86
		Rightwards	27	75.13	7.87



(a) Landscape orientation



(b) Portrait orientation

Figure 3.1: Example of strokes for two different users

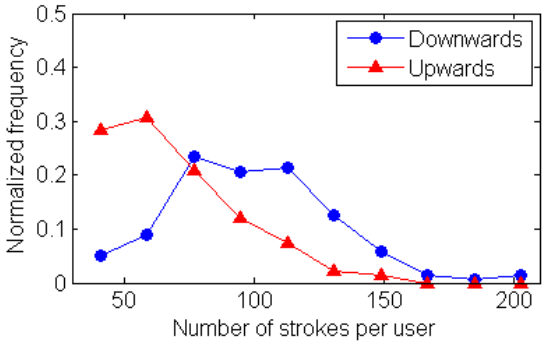
data points. Therefore, given the same sample frequency, horizontal strokes must be performed, in general, faster.

In figure 3.2 the distribution of the number of strokes across the users is depicted. It can be observed that vertical strokes have an initial increase and then a decrease, while horizontal strokes have more uniform distributions, with a greater variation across users.

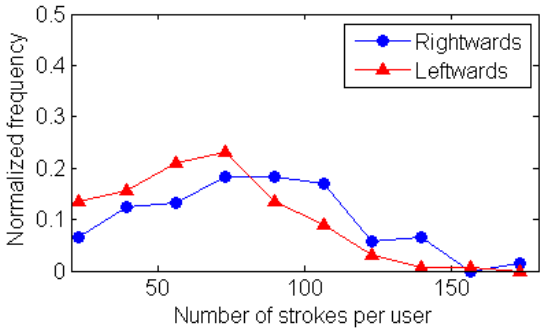
The distribution of the number of points per stroke can be found in figure 3.3. Horizontal strokes have much less data points than vertical ones. The distribution in horizontal strokes is similar for portrait and landscape orientation, with a peak near 7 points per stroke and then a decrease. Downwards strokes' distributions also present similarities, with the mode around 17 points in both orientations. Although there is first an increase and then a descend around 27 points, the similitude in upwards strokes is smaller.

3.1.2 Preprocessing

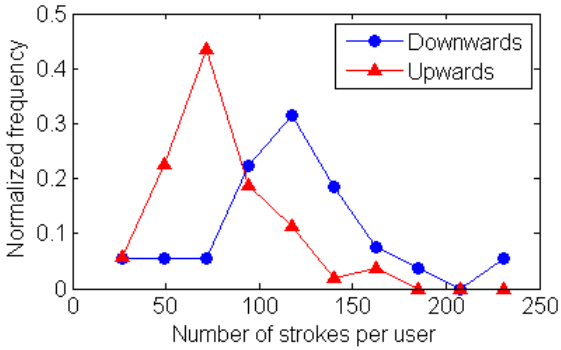
Short strokes of less than five touch points, which probably come from taps on the screen, are considered outliers and discarded. Landscape and portrait data are processed separately, because some features, as can be for example the coordinates from start and endpoints, velocity or duration, may change in the same user depending of the orientation. Therefore, if not divided in two different categories, the intra-class variance for each user augments. The strokes from



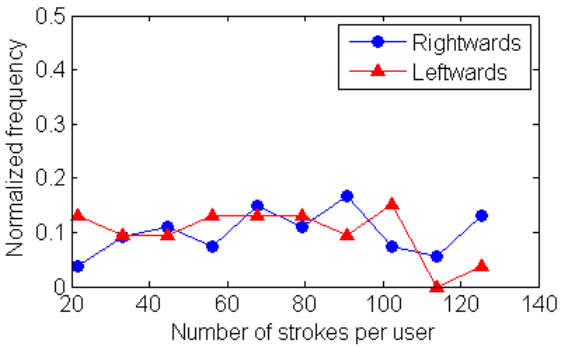
(a) Portrait Vertical



(b) Portrait Horizontal

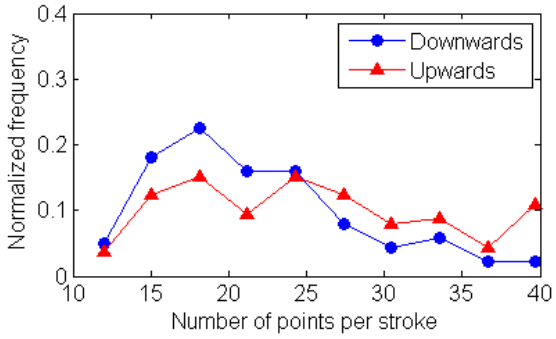


(c) Landscape Vertical

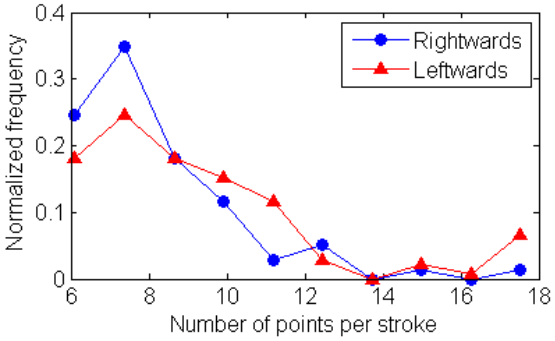


(d) Landscape Horizontal

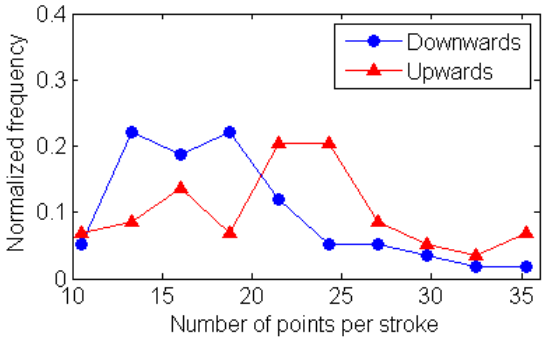
Figure 3.2: Distribution of the number of strokes per user



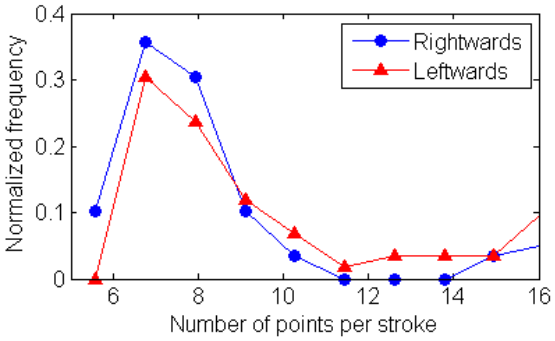
(a) Portrait Vertical



(b) Portrait Horizontal



(c) Landscape Vertical



(d) Landscape Horizontal

Figure 3.3: Distribution of the points per stroke

each orientation are first classified in vertical or horizontal. A stroke is considered vertical if the vertical displacement is bigger than the horizontal one. If the horizontal displacement is bigger than the vertical one, the stoke is considered horizontal. Furthermore, in some experiments, in the vertical class strokes are divided by their direction in upwards or downwards. Following the same procedure, horizontal strokes are divided in rightwards and leftwards.

The classification in vertical and then upwards/downwards strokes, and in horizontal and rightwards/leftwards strokes, is motivated by these gestures being so commonly done that certain habits are developed. Hence, each of them is made in a different way, for example with different fingers, and have their own characteristics that define them. By dividing the strokes in four separate classes we can take the particularities of how each user makes each gesture into account.

Additionally, if a gesture is not correctly divided in different strokes, and it is formed by two different ones, for example, if the user did not rise his or her finger between swipes, it is divided in two different strokes by its inflection point.

3.2 Feature extraction

Following the procedure in [2], after data cleansing, a feature vector for each stroke is calculated. Each feature vector has the characteristics used for describing a particular stroke. Two different feature vectors, proposed in the literature, are used in this work. The first one is a vector used for touch data in [2]. The second one is a vector based on the global feature set used for online signatures in [1]. Additionally, for both feature vectors normalization into the interval (0,1) is computed using tanh-estimators or min-max normalization [14].

3.2.1 Touch data feature set

For each of the strokes in the dataset a 28-dimensional feature vector, previously used in [2], is computed. A vector of velocity and a vector of acceleration are computed for every pair of adjacent points in a stroke, as the first-time and second-time derivative, respectively of the position coordinates. If t, x and y specify the time, the x-coordinate and the y-coordinate of a stroke and k one of its points, the velocity vectors in each coordinate are calculated as follows:

$$v_{x_k} = (x_k - x_{k-1})/(t_k - t_{k-1})$$
$$v_{y_k} = (y_k - y_{k-1})/(t_k - t_{k-1})$$

This results in the following velocity and acceleration vector:

$$v_k = \sqrt{{v_{x_k}}^2 + {v_{y_k}}^2}$$
$$a_k = (v_k - v_{k-1})/(t_k - t_{k-1})$$

For these two vectors, as well as for the pressure and area measurements the mean, the standard deviation, the first quartile, second-quartile and third quartile are calculated. The eight remaining features are:

- The x and y-coordinate of the most extreme points in a stroke. If it is vertical, these points are the uppermost and bottom points. If it is horizontal these points are the most leftward and the most rightward points. Each of the coordinates contributes as one feature.
- The distance between start and end points of a stroke

Table 3.2: Touch data feature vector

	#	Feature	Description
Velocity	**1-5**	Mean, standard deviation, first quartile, second quartile, third quartile	$\bar{v}_n$; σ_{v_n}; $Q_1(v_n)$ $Q_2(v_n)$; $Q_3(v_n)$
Acceleration	**6-10**	Mean, standard deviation, first quartile, second quartile, third quartile	$\bar{a}_n$; σ_{a_n}; $Q_1(a_n)$ $Q_2(a_n)$; $Q_3(a_n)$
Pressure	**11-15**	Mean, standard deviation, first quartile, second quartile, third quartile	$\bar{p}_n$; σ_{p_n}; $Q_1(p_n)$ $Q_2(p_n)$; $Q_3(p_n)$
Area	**16-20**	Mean, standard deviation, first quartile, second quartile, third quartile	$\bar{A}_n$; σ_{A_n}; $Q_1(A_n)$ $Q_2(A_n)$; $Q_3(A_n)$
Position	**21-24**	Extreme points	$(x_1, y_1); (x_2, y_2);$ $(x_3, y_3); (x_4, y_4)$
	25	Sum of distance between adjacent points	$d_n = \sum \sqrt{\Delta y_n^2 + \Delta x_n^2}$
	26	Angle between endpoints	$\theta_n = arctan(\Delta y_n / \Delta x_n)$
	27	Distance between endpoints	$D_n = \sqrt{(y_{end} - y_1)^2 + (x_{end} - x_1)^2}$
Time	**28**	Total duration	$T_n = t_{end} - t_1$

- The angle of the line that joins start and end points of a stroke
- The total duration of the stroke
- The summation of the distance between every pair of adjacent points in the a stroke

Table 3.2 shows a complete list of the 28-dimensional vector.

3.2.2 Signature feature set

This feature set is adapted from the 100-dimensional feature vector used in online signature from [1]. The use of this feature vector is motivated by the fact that in the same way that touch biometrics do, temporal features of gestures made on a surface are extracted in online signatures, as has been stated in the literature [4]. However, signatures are much more sophisticated and hence, need more complex features, so to adapt the feature vector several features that cannot be applied to this problem were deleted, such as those relating to pen-ups and pen-downs, resulting in a 61-dimensional feature vector. Following the classification found in [1], these 61 characteristics can be summed up as follows, with a complete specification in table 3.3:

- Time: related to stroke duration and to the moments in which events such as local maxima and minima took place: 1-10, 12-17
- Velocity and Acceleration: statistics from the velocity and acceleration vectors: 18-42
- Direction: such as the angle between different data points: 45-47
- Geometry: related to the displacement in the x and y-coordinates: 11, 43, 44, 48-61

It should be noted that all features but the pressure and area ones described in section 3.2.1 are present in this feature vector. Thus, eighteen of them can be found in both feature vectors.

Table 3.3: Signature feature vector adapted from [1]. Table adapted from [1]

#	**Feature**	#	**Feature**
1	stroke total duration T_s	**2**	(1st $t(v_{max})$)/T_w
3	$T(v_x > 0)/T_w$	**4**	$T(v_x < 0)/T_w$
5	$T(v_y > 0)/T_w$	**6**	$T(v_y < 0)/T_w$
7	(1st $t(v_{y,max})$)/T_w	**8**	(1st $t(v_{y,min})$)/T_w
9	(1st $t(v_{x,max})$)/T_w	**10**	(1st $t(v_{x,min})$)/T_w
11	$T((dy/dt)/(dx/dt) > 0)$/ $T((dy/dt)/(dx/dt) < 0)$	**12**	T(curvature> $threshold_{curv}$)/T_w
13	(1st $t(x_{max})$)/T_w	**14**	(2nd $t(x_{max})$)/T_w
15	(3rd $t(x_{max})$)/T_w	**16**	(2nd $t(y_{max})$)/T_w
17	(3rd $t(y_{max})$)/T_w	**18**	average velocity $\bar{v}/v_{max}$
19	$N(v_x) = 0$	**20**	$N(v_y) = 0$
21	$\bar{v}/v_{x,max}$	**22**	$\bar{v}/v_{y,max}$
23	(velocity rms v)/v_{max}	**24**	(centripetal acceleration rms a_c)/a_{max}
25	(tangential acceleration rms a_t)/a_{max}	**26**	(acceleration rms a)/a_{max}
27	(integrated abs. centr. acc. a_{ic})/a_{max}	**28**	(velocity correlation $v_{x,y}$)/${v_{max}}^2$
29	standard deviation of v_x	**30**	standard deviation of v_y
31	standard deviation of a_x	**32**	standard deviation of a_y
33	average jerk	**34**	$\overline{J_x}$
35	$\overline{J_y}$	**36**	j_{max}
37	$j_{x,max}$	**38**	$j_{y,max}$
39	j_{rms}	**40**	$t(j_{max})/T_w$
41	$t(j_{x,max})/T_w$	**42**	$t(j_{y,max})/T_w$
43	N(sign changes of dx/dt and dy/dt)	**44**	$T((dx/dt)/(dy/dt) > 0)$/ $T((dx/dt)/(dy/dt) < 0)$
45	θ(initial direction)	**46**	θ(before finger up)
47	θ(finger-down to finger-up)	**48**	$A_{min} = (y_{max} - y_{min})(x_{max} - x_{min})$ / (max distance between points)/A_{min}
49	$(x_{max} - x_{min})\Delta_y/(y_{max} - y_{min})\Delta_x$	**50**	standard deviation of x/Δ_x
51	standard deviation of y/Δ_y	**52**	$(T_w\bar{v})/(x_{max} - x_{min})$
53	$(T_w\bar{v})/(y_{max} - y_{min})$	**54**	$(x_{max} - x_{min})/x_{acquisitionrange}$
55	$(y_{max} - y_{min})/y_{acquisitionrange}$	**56**	$(\bar{x} - x_{min})/\bar{x}$
57	spatial histogram t_1	**58**	spatial histogram t_2
59	spatial histogram t_3	**60**	spatial histogram t_4
61	$(\bar{y} - y_{min})/\bar{y}$		

Chapter 4

Experiments

The experimental protocol and the results obtained are exposed in this chapter. Three systems have been designed: 1) a discriminative one using a SVM classifier, 2) a statistical system based on GMM and UBM, and 3) a system based on the fusion of both previous systems. To conclude, a discussion of which strokes are more discriminative and the possible causes is made.

4.1 Experimental protocol

Users are trained with the data from the first session of the database while data from the second one is used for the test set. Two different experimental protocols have been followed:

- The first protocol follows the procedure used in [2], to obtain a benchmark that serves as a reference to compare the performance with other experiments. Features are normalized to (0-1) range using min-max normalization. 80 randomly chosen strokes from the legitimate user and 80 randomly chosen strokes in total from impostor users are used to train. For the test set, all samples from the legitimate user are used, while each impostor user contributes with 10 randomly chosen strokes. Additionally, 10 feature vectors are averaged to compute a single feature vector using a sliding window. Thus, every feature vector in the test set is derived from 10 different strokes. For each of these feature vectors a similarity score is obtained by comparing it with the stored template.
- The second protocol uses Gaussian Mixture Models for classification and thus only uses the target user's features vectors for training. For testing, all the feature vectors acquired in session 2, from all users, are used. The SFFS algorithm is applied to the 61-dimensional feature set to obtain the best feature vector. These features are normalized using tanh estimators. The system uses full covariance matrices to obtain the UBM models. All data from the training set is used to obtain the UBM. Afterwards, the UBM is adapted to each user, obtaining the user's model λ_C.

To evaluate the performance each system, the Equal Error Rate (EER) is used. This measure is the error rate at which the probability of false acceptance is equal to the probability of false rejection. Thus, the performance improves as the EER decreases. For both procedures, the EER is computed for each user and the mean and standard deviation over all of them is obtained.

4.1.1 Individual feature analysis

The discriminative power is studied for both feature vectors described in chapter 3 (touch data and signature, see tables 3.2 and 3.3 respectively). The performance is studied on the vertical

Table 4.1: Mean EER (%) of each of the features on the 28-dimensional feature vector

Velocity							
$\bar{v}_n$	σ_{v_n}	$Q_1(v_n)$	$Q_2(v_n)$	$Q_3(v_n)$	Mean		
52.19	47.55	50.87	51.41	51.49	50.72		
Acceleration							
$\bar{a}_n$	σ_{a_n}	$Q_1(a_n)$	$Q_2(a_n)$	$Q_3(a_n)$	Mean		
51.86	50.23	49.89	52.55	49.04	50.71		
Pressure							
$\bar{p}_n$	σ_{p_n}	$Q_1(p_n)$	$Q_2(p_n)$	$Q_3(p_n)$	Mean		
50.22	15.14	51.31	50.68	50.49	43.57		
Area							
$\bar{A}_n$	σ_{A_n}	$Q_1(A_n)$	$Q_2(A_n)$	$Q_3(A_n)$	Mean		
51.01	46.61	46.24	49.94	50.46	48.85		
Position							
(x_1, y_1)	(x_2, y_2)	(x_3, y_3)	(x_4, y_4)	d_n	θ_n	D_n	Mean
48.79	49.39	51.04	49.53	51.50	43.56	50.26	49.15
Time							
T_n / Mean							
48.37							

swipes from portrait orientation, which are the most commonly done, and measured with the EER using only one stroke.

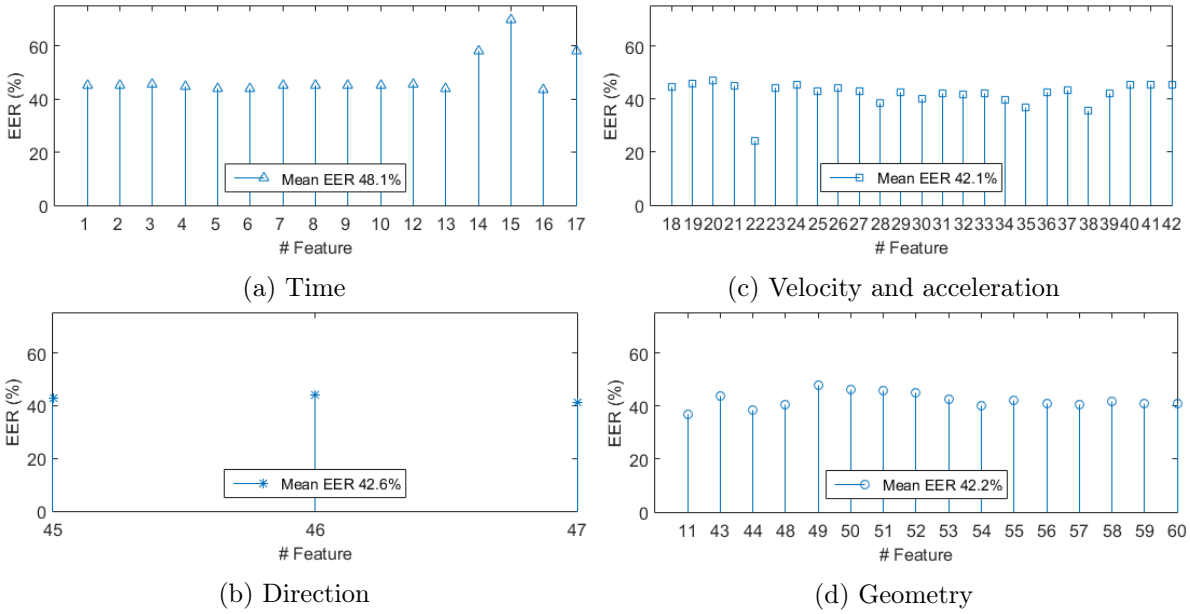
Touch data feature set

The performance of each of the 28 features described in section 3.2.1 is studied. Table 4.1 presents the mean EER of each of the characteristics individually and in the last column the mean of each of the classes. Most of the features have an EER around 50%, which in a two-class verification problem means that the decision is made randomly. In other words, the result is the same as flipping a coin. Only two features present a lower EER. The first one is θ_n, the angle of the swipe, which has a 43.56% EER, slightly lower than the others. σ_{p_n}, the standard deviation of the pressure in a stroke, has the lowest EER, 15.14%. This low EER may be caused by the habit of rising or not the finger off the screen at the end of the stroke, thus having more or less variance, becoming a characteristic feature.

Given the results, it can be concluded that alone the features do not present discriminative power and should be used together. This may be due to the fact that strokes are very simple movements, with only a few data points to obtain information, and as a consequence if only one feature is studied a lot of users do it in the same way and it has by itself low inter-user variability, while combining more features results in more characteristic and discriminative patterns.

Signature feature set

The contribution of each of the features from the 61-dimensional vector described in section 3.2.2 can be found in figure 4.1. This figure presents the features divided by their type and shows both the mean EER and each feature's EER. The results are better than the ones obtained with the 28-dimensional vector, with most EER around 40% instead of 50%. The worst performing features are the time features, while the rest have similar discriminative properties. The most discriminative feature with an EER of 22.4% is number 22, which relates the average velocity in



(a) Time

(c) Velocity and acceleration

(b) Direction

(d) Geometry

Figure 4.1: EER (%) in the different type of features of the 61-dimensional feature vector

the y axis and its maximum velocity. Thus, this feature may discriminative due to some users making steady strokes with the same average velocity, while others change it more. Similar results may be expected in the equivalent feature for the x axis in horizontal strokes.

The results obtained show that alone the features in the 61-dimensional vector are more discriminative than those used in the 28-dimensional vector. A possible cause is that the features in this vector study the x and y axis separately, and therefore can study characteristic habits in each direction and not the result of combining both.

4.2 Feature selection

Feature selection is performed on the 61-dimensional feature set presented in table 3.3 using all users in the database. Vertical strokes in portrait orientation are used with the SFFS algorithm. 80 strokes were randomly selected from the training set for feature selection per user, of which 50 were used for training and 30 for testing. In this case, the user model is obtained modelling a GMM with 3 Gaussian components directly using the user's data (no UBM). In figure 4.2a the evolution of the performance as the feature vector size grows is depicted. It can be observed that there is a big difference in the system's performance between selecting four and five features. Adding further features does not seem to obtain better performance, probably because they don't complement well the others and don't add more discriminative information.

To verify the results obtained by the SFFS algorithm the performance with the statistical system including UBM is studied, ensuring as well that the features selected work well with this system. The relevance factor used is $r = 5.5$ with 3 GMM components. 20 training samples are used to adapt the UBM to the user model. Figure 4.2b shows the results obtained when using only one stroke to obtain each score. The EER obtained with the SFFS is not improved. This can be due to using more samples in the testing set for verification, which may cause that more similar strokes are compared with the legitimate ones. Nevertheless, the optimal feature size is still five features and as happened with the SFFS algorithm, the results don't get better when adding more features. It should also be mentioned, that in both the verification and the algorithm results, there is an approximated EER fall of 10% between using four and five features.

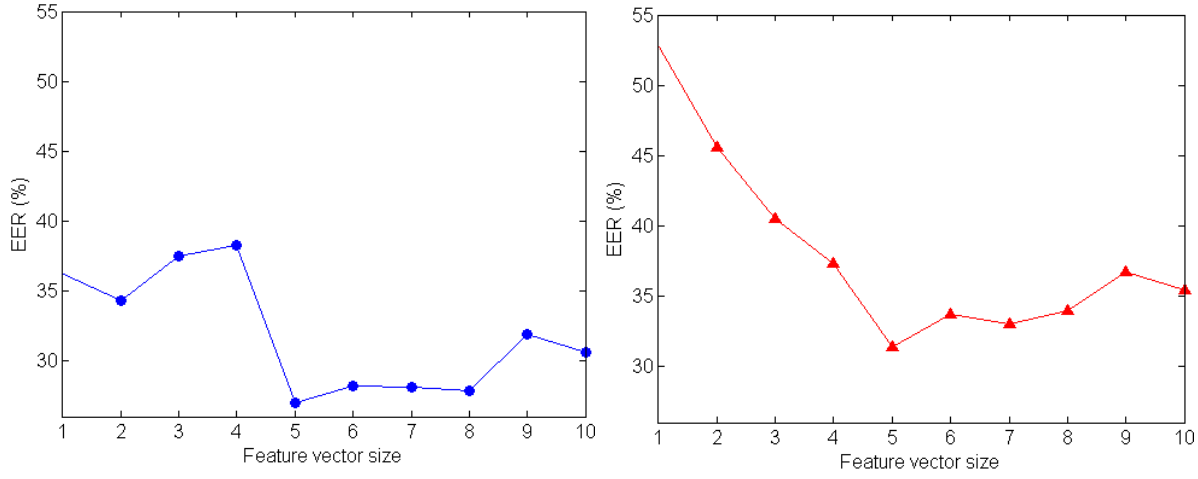


Figure 4.2: (a) Performance depending on the size of the optimal feature set selected by the SFFS algorithm (GMM without UBM). (b) Performance using all the users in the database in the statistical system (GMM with UBM)

The best features chosen by this algorithm are depicted in table 4.2. The features relating to the x-axis have been changed for horizontal strokes to their equivalents in the y-axis. Of these five features, there is a direction feature, an acceleration feature and the other three are geometry related. According to the SFFS algorithm these are the five features that together are more discriminative. Despite the small dimension of the resulting feature vector, it should be noted that the gestures used for classification they can be easily described due to their simplicity.

Two features from the best performing two-dimensional vector found by SFFS will be used to represent the GMM and see how well they adjust to the data. The first of these features will. be for all the gestures, the angle between the first and the last point. However, the second feature will change depending if the swipe is vertical or horizontal. For vertical strokes the normalized maximum deviation in x will be used, while for horizontal ones the normalized maximum deviation in y will be used. These features' physical meaning is depicted in figure 4.3.

4.3 Experimental results

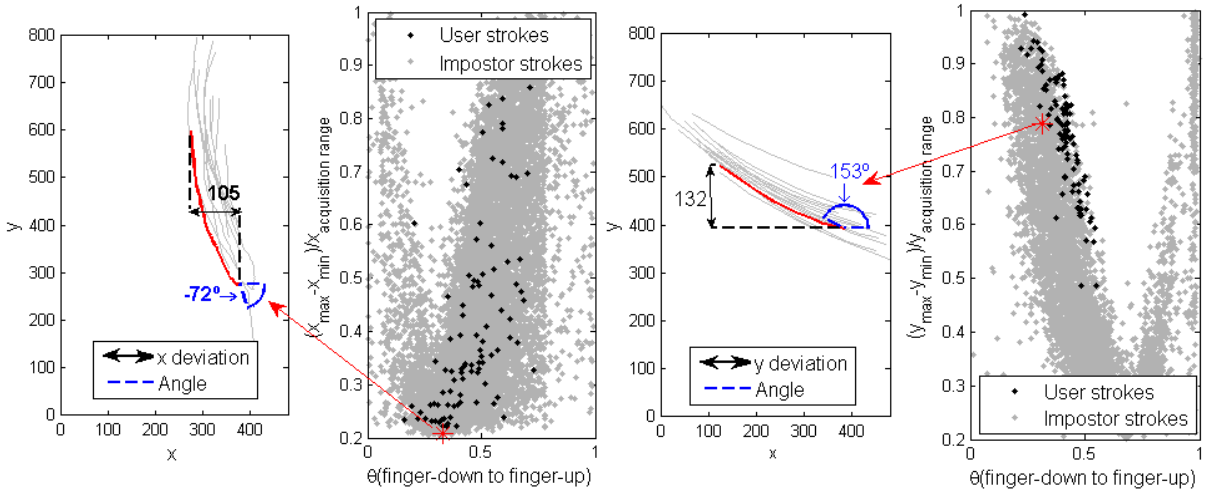
4.3.1 Performance of the discriminative system

Benchmark obtained following protocol of the literature

To obtain a benchmark that can be used as a reference to compare our performance with the literature, the procedure proposed in [2] has been implemented. Following the same steps, horizontal and vertical swipes have not been divided in further classes depending on their direction.

Table 4.2: Optimal feature set selected by the SFFS algorithm

Scenario	Best performing features
Vertical	θ(finger-down to finger-up), $(x_{max} - x_{min})/x_{acquisitionrange}$, standard deviation of a_x, $(\bar{x} - x_{min})/\bar{x}$, $(y_{max} - y_{min}/y_{acquisitionrange}$
Horizontal	θ(finger-down to finger-up), $(y_{max} - y_{min})/y_{acquisitionrange}$, standard deviation of a_y, $(\bar{y} - y_{min})/\bar{y}$, $(x_{max} - x_{min})/x_{acquisitionrange}$



(a) Vertical stroke

(b) Horizontal stroke

Figure 4.3: Physical meaning of the features used to represent the data. Points in grey show impostors data points, while the legitimate user data is in black

Authentication in this system is performed using SVM. The regularization parameter C and the Kernel's variance σ^2 are tuned to each user, choosing from a range of values the one that gives back the best performance.

In table 4.3 the results obtained are depicted. This table shows both the mean and the standard deviation of the EER obtained for all users separating landscape and portrait data and horizontal and vertical strokes. The results show that the lowest EER is obtained for horizontal swipes in landscape orientation, so these swipes are more discriminative and users possibly execute them in a more characteristic manner. Landscape orientation results are better than portrait ones, which may be related to the fact that less users have made this kind of strokes, so users who do are very used to swiping with this orientation, and have more distinctive habits.

The results obtained here and shown in table 4.3 are approximately from 3% to almost 10% worse than the ones showed in the reference article [2] following the same protocol, depicted in table 4.4. However, it should also be noted that the same implementation for the SVM classifier has not been used. Anyway, we have interpreted our implementation as close as possible to the details given in [2].

Performance obtained changing the reference protocol

Additionally, three changes in the reference protocol are made to test the SVM system. First, in order to be less affected by outliers tanh estimators are used to normalize the features to (0-1) range. Second, instead of using only 10 samples per impostor in testing, all the data from the second session is used as the test set. Last, rather than averaging 10 features vectors together, the similarity score obtained by 10 test samples is averaged together. The strokes are further divided based on their direction in upwards, downwards, leftwards and rightwards. The motivation is that despite a gesture being both vertical, upwards and downwards strokes are performed differently, with their own characteristic features, in the same way that one does not walk forward and backwards the same way. Figure 4.4 depicts the resulting framework used in this and the following sections.

The performance obtained after these changes can be found in table 4.5. As before, landscape strokes show better results than portrait. Rightwards strokes in both orientations and leftwards

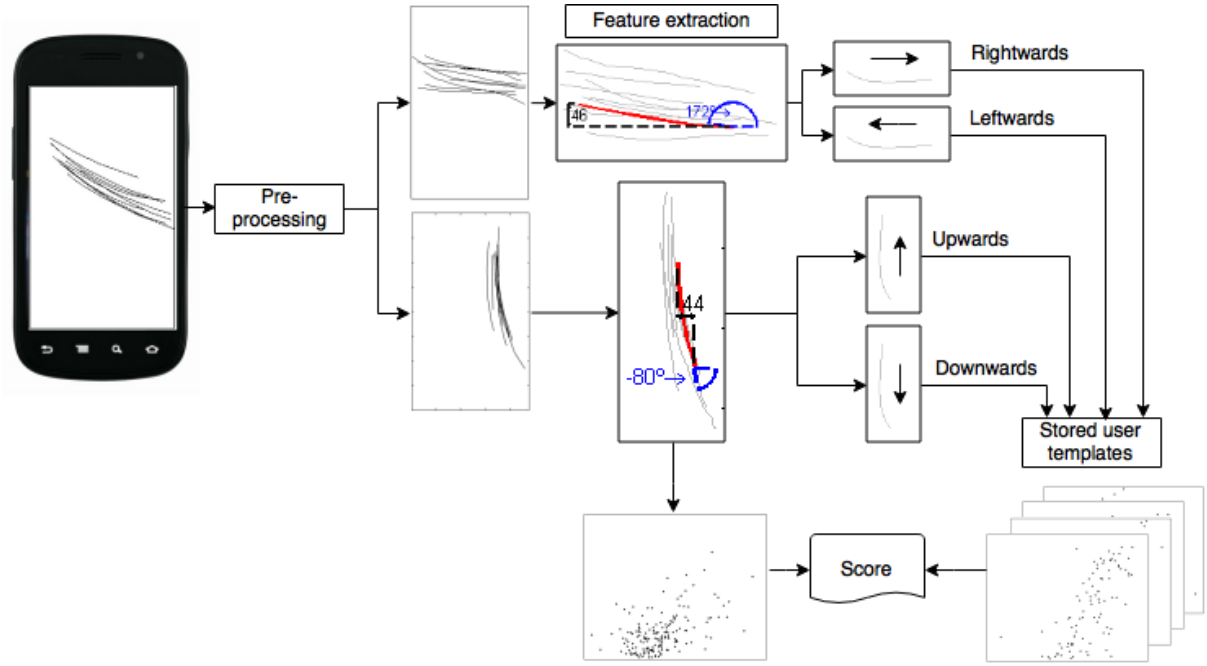


Figure 4.4: Experimental framework

Table 4.3: EER mean and standard deviation obtained for the 28-feature vector using SVM

	Portrait		Landscape	
	Vertical	**Horizontal**	**Vertical**	**Horizontal**
Mean EER (%)	25.44	23.85	21.33	17.49
Standard deviation	11.65	11.99	9.52	11.22

Table 4.4: Performance in terms of EER obtained in [2] with the same protocol

	Portrait		Landscape	
	Vertical	**Horizontal**	**Vertical**	**Horizontal**
Mean EER (%)	15.7	18.0	13.1	14.7
Standard deviation	13.6	13.6	13.1	9.8

Table 4.5: Performance of the SVM system changing the reference protocol

			EER	
			Mean (%)	**Standard deviation**
Portrait	**Vertical**	**Upwards**	16.05	9.86
		Downwards	14.73	8.85
	Horizontal	**Leftwards**	16.06	12.33
		Rightwards	9.95	6.89
Landscape	**Vertical**	**Upwards**	9.95	7.13
		Downwards	17.11	9.17
	Horizontal	**Leftwards**	8.12	6.88
		Rightwards	8.10	6.93

and upwards strokes in landscape orientation show the best EER. Thus, they hold more discriminative information than the other strokes. Compared to the results obtained with the benchmark shown in table 4.3, the performance has significantly improved, decreasing around 10% in all classes. Furthermore, in contrast with the results announced in [2], the horizontal strokes in both portrait and landscape orientation show a better performance, while vertical strokes show a similar result.

4.3.2 Performance of the statistical system

An evaluation of the performance of the statistical system based on GMM adaptation is carried out in this section. In the first place, an evaluation analyzing if the data is well represented is made. Afterwards, the UBM model and its adaptation to the user data is studied. This is followed by an analysis regarding the number of Gaussian components, relevance parameter r and the number of training samples to adapt the UBM. Lastly, the UBM adaptation system and a system using GMM directly to obtain the user's model are compared. In all these cases the performance reported is the EER obtained when computing the mean of ten scores, as it has been proven in the literature [2, 4, 6, 10] that blocks of strokes work better.

Representation of the data with Gaussian Mixture Models

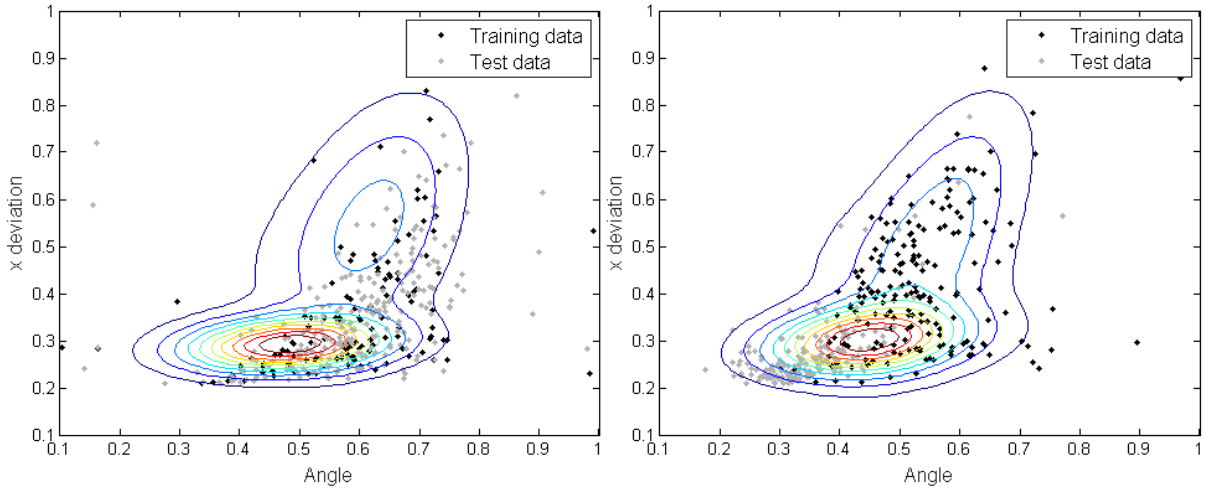
First, a study on whether the data can be represented by GMM is performed. Three Gaussian components are chosen to model the GMM. Figure 4.5a shows the GMM estimated for a user with one of the best performances. Most of the data is gathered around the bottom, although some data points are situated above. Then, to model this distribution the GMM has one of the Gaussian components located on top of most of the data on the bottom, while the other two reach out to the outliers above.

In figure 4.5b can be found a representation of the GMM estimated for a different user, whose performance is worse. Nevertheless, the estimated GMM in both users is very similar. The disparity in performance is caused by the fact that the training and test data for the first user overlaps and is condensed around the same area. Hence, the GMM estimation gives back a good result. On the other hand, in the second user, whose performance is much worse, the training data is mostly around the top Gaussian components and the test data is on the bottom. This disparity on the location of the data from the training and test data results in a bad estimation and a bad performance.

These results help to anticipate situations where this system is more limited. When there is a big disparity between the location of the training and the test data, the performance will be worse. Likewise, if the locations are similar the performance will be better.

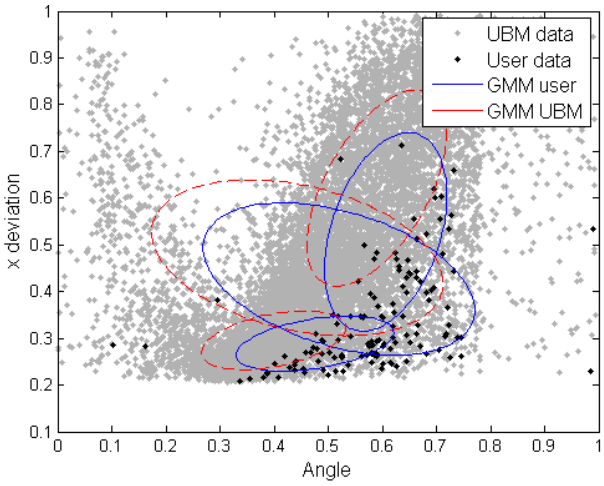
GMM adaptation

To study how the GMM adaptation represents user's data a system with three Gaussian components and relevance factor $r = 5.5$ is used. 40 strokes are used to adapt the UBM. In figure 4.6a the three Gaussian components of the UBM are plotted with their adaptation to the user's data. Each of the Gaussian components has moved towards the user data to represent it best and is located on top of it. Figures 4.6b and 4.6c show respectively, the UBM and its adaptation, obtained by calculating a weighted sum of each of the Gaussian components. The results show how the UBM covers most users' data, while the specific GMM model for the user is located over the user's data whilst still covering areas that have a lot of data from other users. Even though there were no data points in the user's training set in these areas, they are more likely

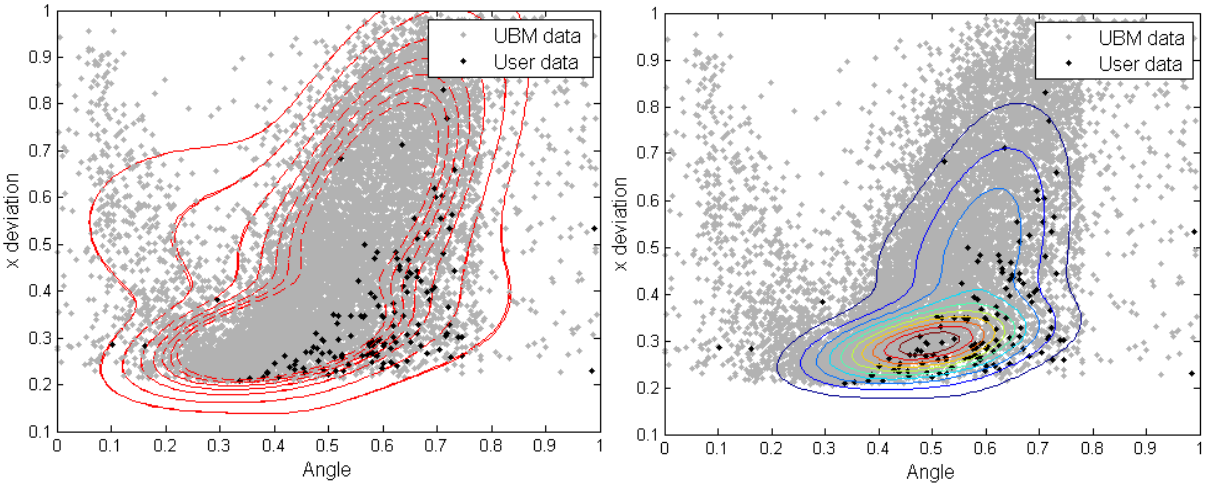


(a) (b)

Figure 4.5: Example of the GMM components estimated for two different users overlapped with the data used for train (black) and test (gray). (a) Depicts a user with good performance, while (b) depicts a user with bad performance

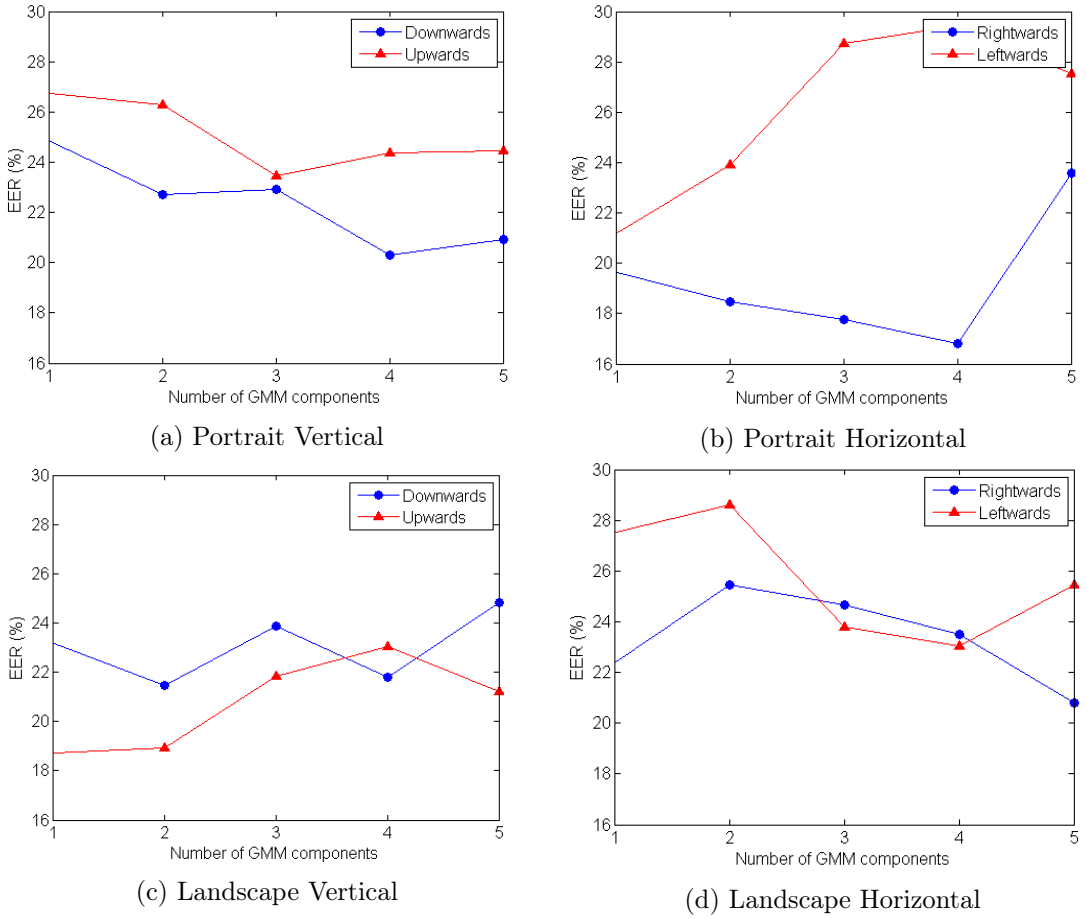


(a) GMM components of the user and UBM models



(b) UBM model (c) User's model adapted from the UBM

Figure 4.6: Example of the UBM model and its adaptation to obtain the user's model



(a) Portrait Vertical

(b) Portrait Horizontal

(c) Landscape Vertical

(d) Landscape Horizontal

Figure 4.7: Effect on the performance of the number of Gaussian components

to fall there, given the accumulation from other users. Nevertheless, the Gaussian component covering those areas has a smaller weight than in the UBM.

To sum up, the results shown prove that the UBM obtained is a general user model, which covers most of the users' data. The adaptation is capable of obtaining a user specific model from this UBM that successfully represents the user's behaviour using a small number of samples.

GMM configuration

The effect on the performance of the number N of GMM components with a fixed relevance factor $r = 5.5$ and 20 training samples per user to adapt the UBM is first studied. In figure 4.7 it is shown that the number of components that result in the best performance changes depending on the type of the stroke. In most cases the performance gets better as the number of components grows. Rightwards strokes perform better for both portrait and landscape orientation than leftwards strokes, respectively. The same can be said about downwards strokes in portrait orientation. This means that downwards strokes in portrait orientation and rightwards strokes are more distinctive. Upwards strokes in landscape orientation and leftwards strokes in portrait orientation obtain a worse performance as the number of GMM components grow. This may be because their data points are very condensed in one only cluster, and when using more components the models represent outliers that do not show the general behaviour of the user.

In landscape orientation, downwards strokes perform worse than upwards and the EER shows a saw-shaped behaviour with better performance with an even number of GMM components.

The most probable cause is that the data is grouped in two clusters and thus an even number of Gaussian components represents the data well while an odd number does not.

As the best result in general is obtained with 4 Gaussian components, table 4.6 presents a summary of the results in that case. This table shows similar performance to the one obtained in the benchmark, whose results are depicted in table 4.3. However, it is worth mentioning that in some cases, i.e. landscape upwards, the result would be better with just one GMM component.

Afterwards, the relevance factor r effect in the adaptation is studied. In figure 4.8 the mean EER across all users in the system for strokes in portrait orientation is depicted as a function of r. It can be observed that as r increases, the EER has a slight initial descend, with some rises and falls, until it stabilizes. This can be caused by the model becoming more general as r increases, thus being less adapted to the user's specific characteristics. However, it should also be noted that even with a large r the adapted model still holds enough user specific information to be discriminative. As was also observed when changing the number of GMM components, downwards and rightwards strokes perform better, respectively, than upwards and leftwards.

The performance for strokes in landscape orientation can be found in figure 4.9. In horizontal strokes the same behaviour as with the portrait oriented strokes can be found: there is a steep decrease and then the EER stabilizes. On the other hand, the vertical strokes have an initial decrease and then increase again when r is big. A possible explanation is that vertical strokes hold less user-specific information, and as r increases the model becomes more general and this information is less present. Thus, the model is less discriminant.

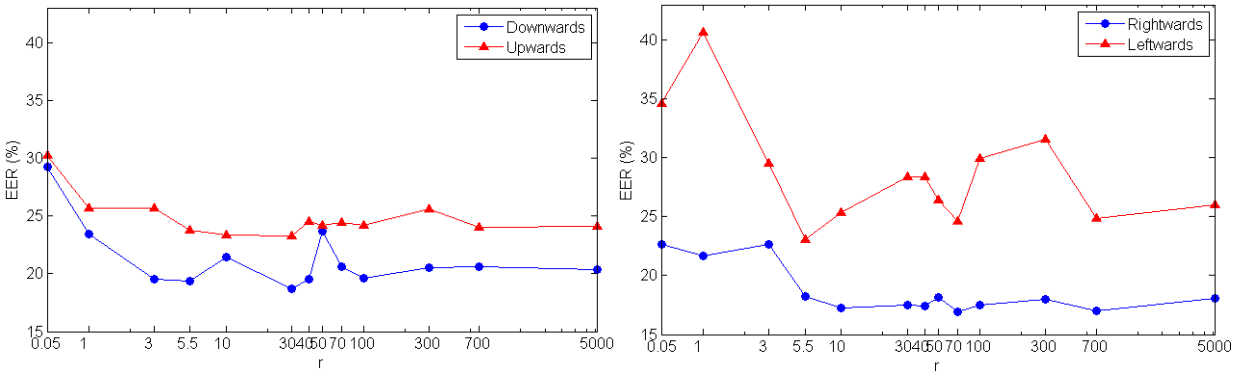
The best performing scenarios seem to be when $r = 5.5$ and with $r = 30$. These scenarios mean EER and their standard deviation are depicted in table 4.7. The results show that the most discriminative strokes are portrait downwards and rightwards and landscape's rightwards and upwards. This behaviour is the same as when studying the number of GMM components and confirms that these gestures are more discriminative. These results improve the benchmark.

Lastly, a study on the number of training samples needed to adapt the UBM is performed. The other parameters used are the best performing in the previous cases, 4 GMM components and a relevance factor $r = 30$.

For portrait strokes, the system performance as a function of the number of training samples can be found on figures 4.10a and 4.10b. It can be observed that in the downwards and rightwards swipes the EER gets lower until after the initial descent it stabilizes around 20 strokes. These types of strokes are the best performing ones, and hold more discriminative information. Leftwards strokes show an unexpected behaviour, with worse performance as the number of samples grow, similar at the one found when changing the number of GMM components depicted in figure 4.7. This suggests that the information is very condensed, with most data points in only one cluster. Choosing 10 or 20 training samples probably uses outliers to adapt the UBM, therefore resulting in a bad performance. On the other hand, upwards strokes show the opposite behaviour, performance improves when using more strokes, but does not stabilize. This may be caused to the data being dispersed and thus to have a representative sample set, a lot of training samples have to be used.

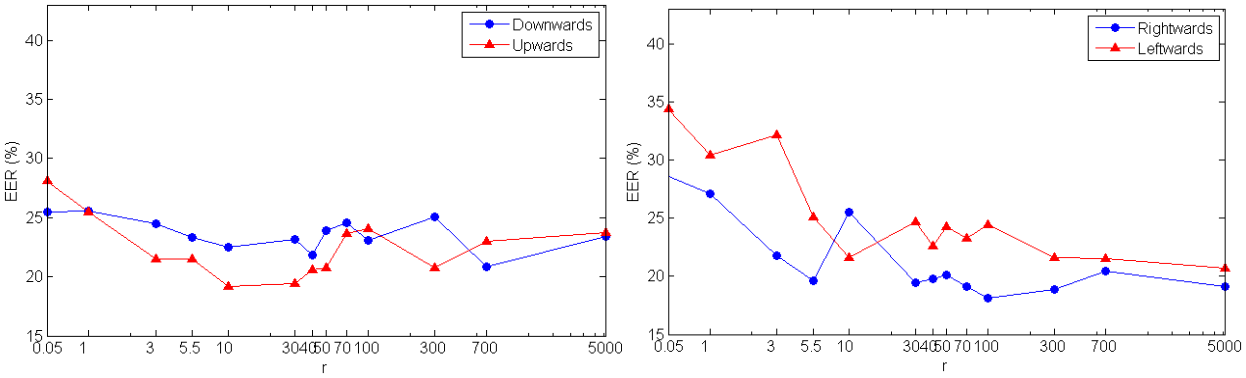
In landscape orientation, the EER obtained as a function of the number of training samples is represented in figures 4.10c and 4.10d. In this case, all kinds of strokes show a steep inclination and after twenty training samples a stabilization. As it has been observed before the best performing strokes and therefore, the most discriminative, are the upwards and rightwards.

Seeing as its the point in which all kind of strokes obtain the best EER, the performance obtained for 40 strokes is summarized in table 4.8. Horizontal strokes in both phone orientations show a better performance, which may indicate that these strokes are more discriminative. Compared to the benchmark obtained with SVM, whose results are presented in table 4.3, the



(a) Vertical (b) Horizontal

Figure 4.8: System performance in portrait orientation as a function of r



(a) Vertical (b) Horizontal

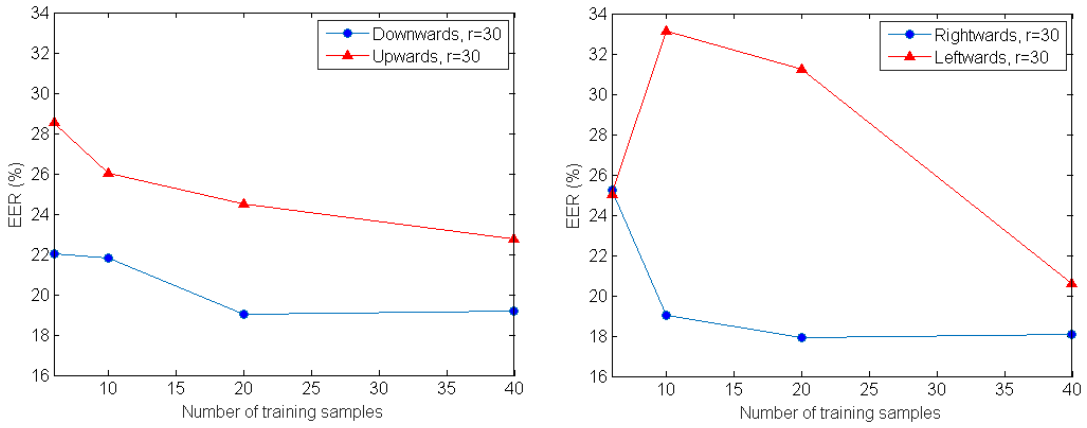
Figure 4.9: System performance in landscape orientation as a function of r

Table 4.6: Statistical system performance with 4 Gaussian components

			EER	
			Mean (%)	**Standard deviation**
Portrait	**Vertical**	**Upwards**	24.37	11.13
		Downwards	20.31	7.40
	Horizontal	**Leftwards**	29.42	9.51
		Rightwards	16.81	8.35
Landscape	**Vertical**	**Upwards**	23.06	7.28
		Downwards	21.79	8.15
	Horizontal	**Leftwards**	23.04	9.95
		Rightwards	23.48	11.34

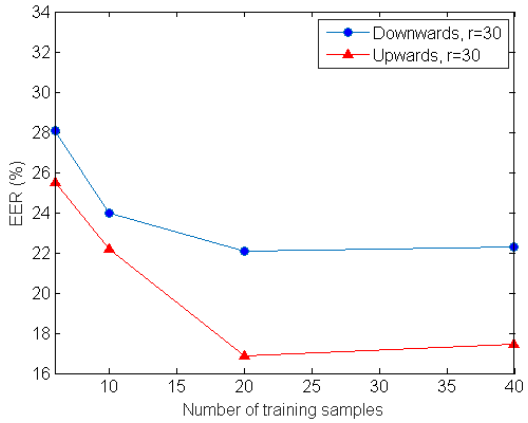
Table 4.7: Statistical system performance with $r = 5.5$ and $r = 30$

			EER			
			Mean (%)		**Standard deviation**	
			$r = 5.5$	$r = 30$	$r = 5.5$	$r = 30$
Portrait	**Vertical**	**Upwards**	23.79	23.23	10.35	10.89
		Downwards	19.38	18.73	7.45	7.20
	Horizontal	**Leftwards**	23.03	28.42	8.70	9.39
		Rightwards	18.25	17.52	9.28	8.96
Landscape	**Vertical**	**Upwards**	21.52	19.42	7.38	7.09
		Downwards	23.35	23.18	10.19	9.12
	Horizontal	**Leftwards**	25.11	24.71	10.05	8.15
		Rightwards	19.65	19.44	10.28	10.73

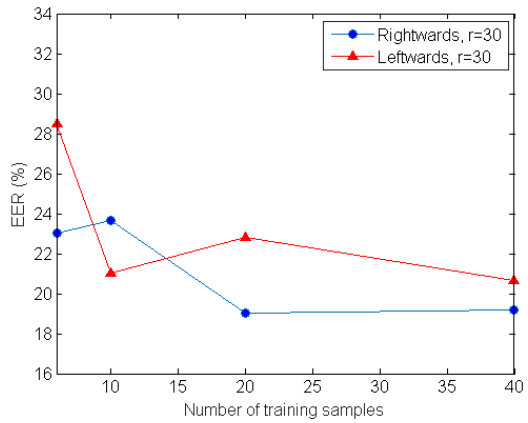


(a) Portrait Vertical

(b) Portrait Horizontal



(c) Landscape Vertical



(d) Landscape Horizontal

Figure 4.10: System performance as a function of the number of training samples

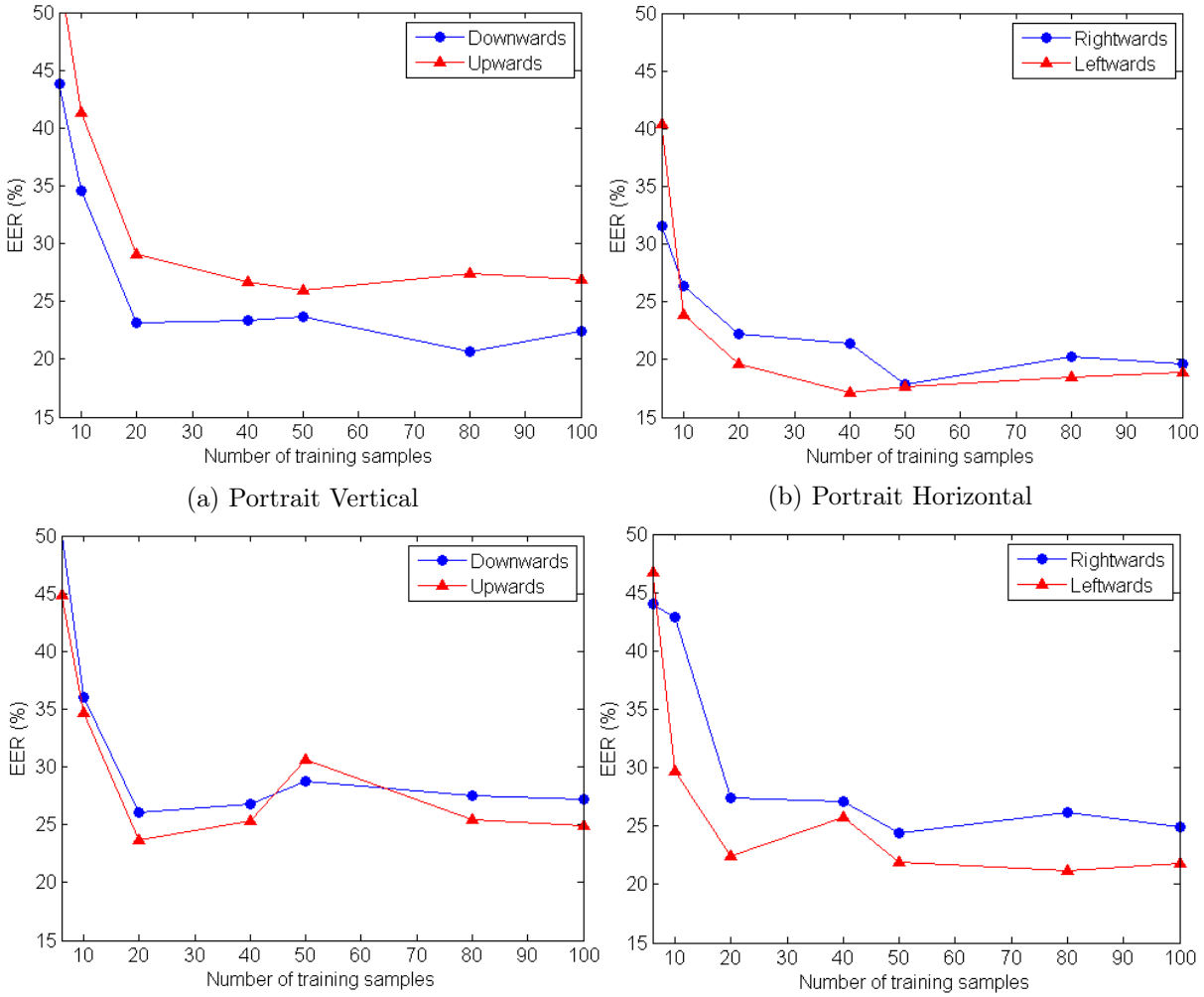
results have improved, with an EER around 20% in all cases. Nevertheless, compared to the results for the discriminative system in table 4.5, the performance is slightly worse. It is worth emphasizing that, for landscape orientation in the statistical system as a smaller number of training samples are needed more users can be used and more comparisons are made. Hence, the result is more reliable.

Performance study using GMM without UBM adaptation

In this section, of the performance using GMM to obtain the user model, without UBM adaptation is studied. This system computes the final score by comparing the score obtained with each user's GMM against the general UBM model. The UBM model is only trained once and is used for all users, while a GMM is trained for each user.

Figure 4.11 depicts the mean EER using 4 GMM components for an increasing number of training samples. These results show that, as the number of training samples grows, the estimation of the user model improves and the EER descends. After an initial steep decrease, in all cases the performance stabilizes. Unlike in previous cases, rightwards swipes perform worse than leftwards. This shows that leftwards strokes are better described when using GMM, maybe because users tend to differ more from the general UBM model and adapt worse.

In the system with GMM adaptation, when the relevance factor r was very small, the Gaussian components are adapted without using the information of the UBM. Therefore, the situation is the same as when the user's model is estimated using only the GMM. For portrait downwards



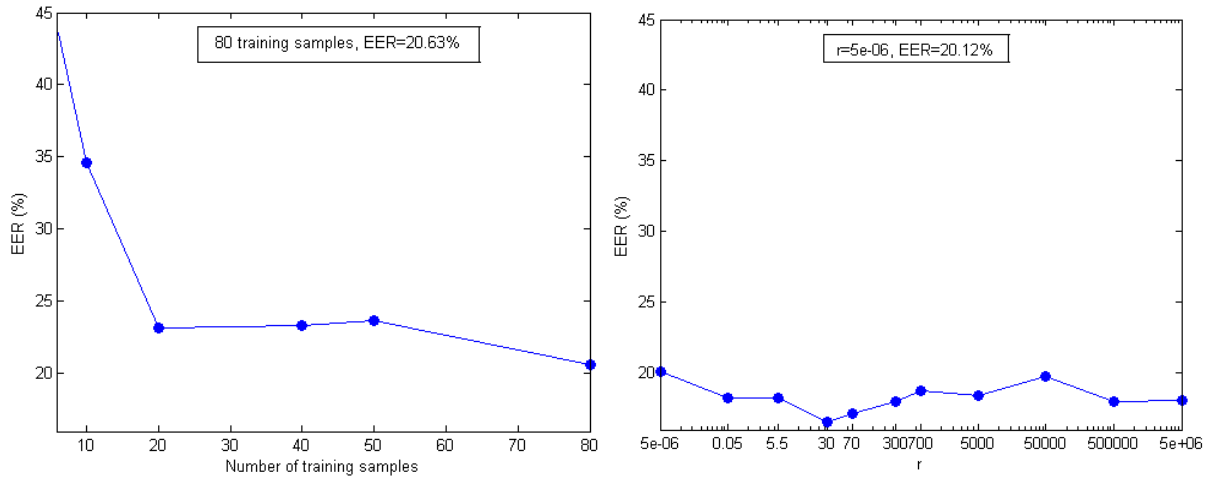
(a) Portrait Vertical

(b) Portrait Horizontal

(c) Landscape Vertical

(d) Landscape Horizontal

Figure 4.11: Effect on the performance of changing the number of training samples without using adaptation to obtain the user's model



(a)

(b)

Figure 4.12: Performance for downwards strokes in portrait orientation obtained modelling the user (a) without adaptation changing the number of training samples and (b) with adaptation for different relevance factors

Table 4.8: GMM with UBM adaptation system performance with 40 training samples

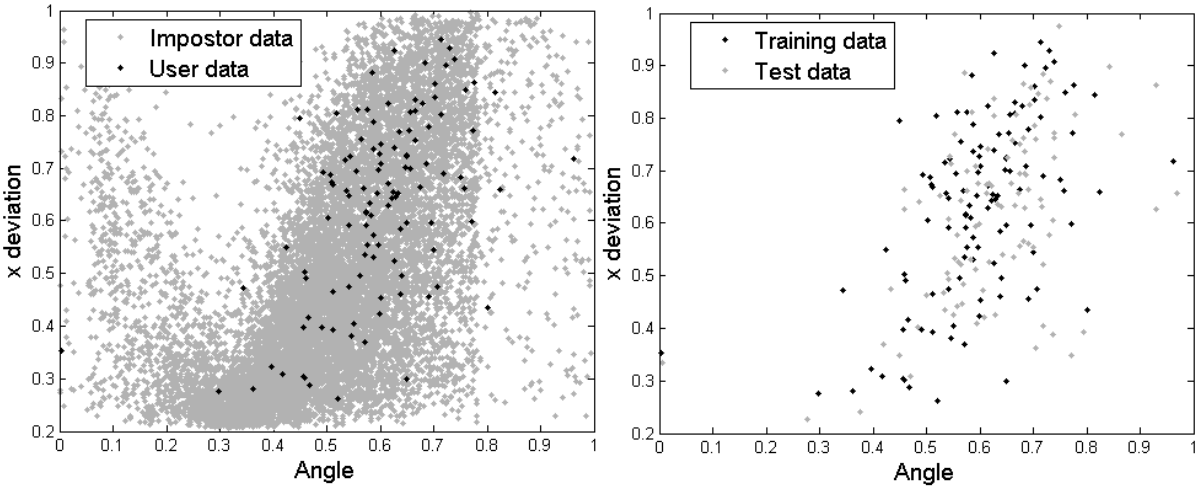
			EER	
			Mean (%)	**Standard deviation**
Portrait	**Vertical**	**Upwards**	22.77	11.24
		Downwards	19.21	8.25
	Horizontal	**Leftwards**	20.60	12.03
		Rightwards	18.09	9.31
Landscape	**Vertical**	**Upwards**	17.45	6.40
		Downwards	22.29	8.94
	Horizontal	**Leftwards**	20.67	10.43
		Rightwards	19.19	10.04

strokes, in figure 4.12 the local system using GMM without adaptation to estimate the user's model is concatenated with the results using the UBM adapted system with 80 training samples per user. It can be observed that the EER obtained with 80 training samples in the GMM user model and with a small r is the same. It should also be noted that the UBM system for the same number of strokes performs better, with a minimum EER of around 16% when $r = 30$.

4.3.3 Comparison of the statistical and the discriminative system

The statistical system classifies users modelling their behaviour and how their data is distributed, while the discriminate system only tries to separate the data without taking into account how it is distributed. Despite the fact that the discriminative system presents a slightly better performance than the statistical, one of the most important limitations it presents is not being able to authenticate users who, although they are stable across sessions, have a big variability. Figure 4.13 shows the typical behaviour of one of these users, who obtain an EER around 50% in the SVM system. It can be observed that the train and test data are distributed similarly. Therefore, it's stable, but it has great variability and is not so condensed as other users' data.

Using the same 80 training samples for both systems, the performance across the 10% worse users in the SVM system is compared with the performance in the system based in GMM adaptation. As very few users swiped the screen in landscape orientation, 10% of these users are



(a) User's data compared to other users

(b) User's data in the training and test set

Figure 4.13: Typical behaviour in users with an EER around 50% in the SVM system

Table 4.9: Comparison of the EER in the two systems for the 10% worst performing users in the discriminative system

			EER	
			SVM	**GMM adaptation**
Portrait	**Vertical**	**Upwards**	57.76	32.56
		Downwards	56.14	25.93
	Horizontal	**Leftwards**	41.75	20.07
		Rightwards	53.91	30.15

Table 4.10: Performance in the SVM system with 40 strokes

			EER	
			Mean (%)	**Standard deviation**
Portrait	**Vertical**	**Upwards**	23.98	17.33
		Downwards	22.93	15.84
	Horizontal	**Leftwards**	21.61	17.66
		Rightwards	22.94	18.26
Landscape	**Vertical**	**Upwards**	14.72	12.37
		Downwards	16.11	10.75
	Horizontal	**Leftwards**	13.22	11.30
		Rightwards	11.80	10.03

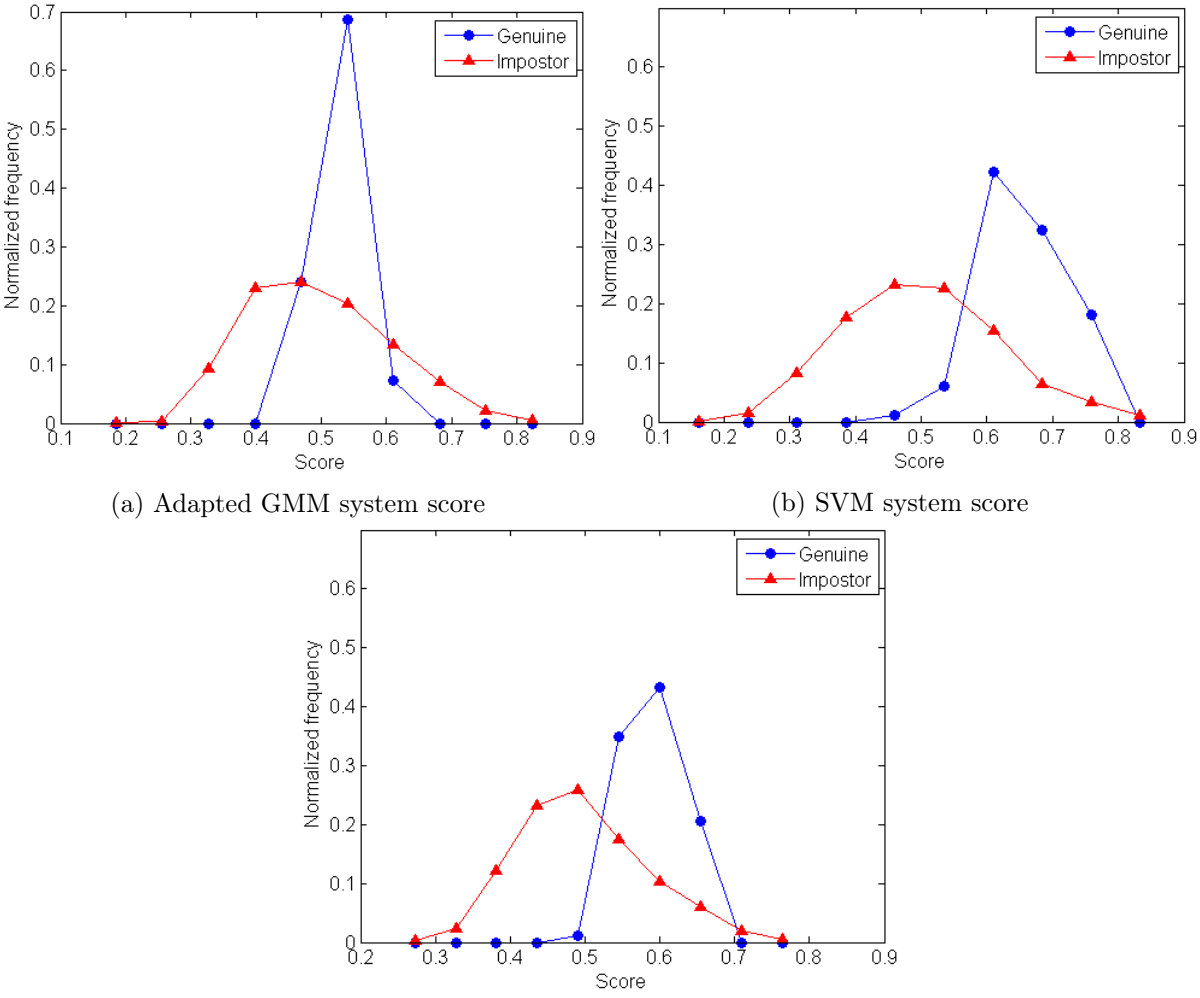
not representative. Hence, only data from portrait orientation is used in this comparison. Table 4.9 depicts the results obtained. It can be observed that, while worse than the average for the statistical system, the performance is much better than in the discriminative system. Therefore, the statistical system is capable of authenticating users that cannot be authenticated with the discriminate system. The reason is that the statistical system models user behaviour, instead of just trying to separate users.

Another limitation of the SVM system is that it needs a high number of training samples. In table 4.10 can be observed the mean EER in the SVM system using only 40 training samples to train. Compared to the results with 80 training samples depicted in table 4.5, the performance deteriorates, while in the adapted GMM system, as could be observed in figure 4.10, the performance is not so affected by the number of training samples. In a real case scenario, where the user's model has to be obtained as fast as possible to protect the device, the time needed to obtain the training samples for each kind of operation needs, thus, to be as low as possible.

4.3.4 Fusion of the discriminative and the statistical systems

Taking under consideration that each of the systems focuses on different aspects of the strokes, a system based on their fusion is proposed. Hence, the information obtained in the previous experiments is merged. The fusion is done by computing the score in each system separately. Scores are normalized using tanh estimators to (0-1) range and then averaged together. An example of the scores obtained in each of the systems and the final score of the fusion can be found in figure 4.14. It can be observed that the scores present Gaussian-shaped distributions and that the genuine and impostor scores overlap, so they are not clearly separable.

As in the previous experiments, 10 scores of the final result are averaged together to use blocks of strokes. The SVM system used is the one developed varying the protocol from [2] described in section 4.3.1 and uses the 28-dimensional feature vector, while the other system uses 4 GMM components, with $r = 30$ and the 5-dimensional feature vector obtained with SFFS.



(a) Adapted GMM system score

(b) SVM system score

(c) Fusion system score

Figure 4.14: Scores in the system based on the fusion of the adapted GMM and the SVM systems

Table 4.11: Performance by combining the SVM and the adapted GMM systems

			EER	
			Mean (%)	**Standard deviation**
Portrait	**Vertical**	**Upwards**	16.79	13.94
		Downwards	14.46	11.75
	Horizontal	**Leftwards**	12.18	13.01
		Rightwards	14.89	13.12
Landscape	**Vertical**	**Upwards**	12.16	17.75
		Downwards	10.86	9.28
	Horizontal	**Leftwards**	8.35	7.40
		Rightwards	10.32	9.56

In training, the same 80 samples from the legitimate user are used in both systems and in the SVM system 8 users chosen randomly contribute with 10 samples each for the negative class.

An overview of the results obtained can be found in table 4.11. It shows that this system performance in all cases is better than in the statistical system. Nevertheless, the EER obtained in the discriminative system is not improved in rightwards strokes, both in portrait and landscape orientations, as well as in landscape upwards. The overall performance has improved compared to the other systems and is also better than the performance obtained in the reference article [2]. It is worth noting that while rightwards gestures achieved a better result than leftwards swipes in previous experiments, they perform worse in this case. A possible explanation is that for this kind of gestures the two systems don't merge well together and obtain very different scores.

4.3.5 Discussion of the performance across touch operations

The best performing gestures in portrait orientation have been, in most cases, the downwards and rightwards strokes over the upwards and leftwards strokes. This means that they hold more discriminative information than the others. Therefore, users swipe the screen in more particular patterns in these gestures. A possible reason for may be that they are performed more often and are more stable. Additionally, it should be noted, that the horizontal strokes perform better than the vertical ones. The cause may be related to the fact that horizontal strokes are shorter than vertical ones. Thus, horizontal gestures have less degrees of freedom and users tend to show a more stable behaviour.

On the other hand, in landscape orientation, all strokes obtain a similar performance, with horizontal strokes obtaining slightly better performance than vertical ones. Considering that horizontal strokes also perform better in portrait orientation, they probably hold more discriminative information. As happened in portrait orientation, this may be due to being shorter and hence, having less variability. However, as the number of users is much smaller than in other touch operations the results are also less trustworthy. It should be emphasized that, although their performance when fusing the systems is better, downwards strokes obtained a bad performance both in the statistical and discriminative systems. A possible cause may be that they do not hold as much information as other gestures and each of the systems focuses on only a part of this information. For example, the signature feature set does not use pressure information while the 28-dimensional feature set does. The two systems may complement each other information and obtain a better performance.

Less users swiped the screen in landscape orientation compared to portrait orientation. Therefore, it can be hypothesized that users who make this gestures are very used to doing them and have developed stable habits. This may be one of the reasons why in landscape orientation the overall performance is better.

Chapter 5

Conclusions and future work

5.1 Conclusions

This work has analysed the possibility of using gestures made during the usual interaction with a touchscreen to authenticate a user. The biggest problem for touch biometrics is the high-variability of touch behaviour within the same user, which is ratified by the study of the discriminative power of the features sets. This analysis shows that features in the selected 61-dimensional vector (one of the best feature sets form the signature biometrics literature [1]) show greater individual discriminative power than those in the 28-dimensional vector from the selected reference work [2] from which we borrow our experimental protocol and database. This is probably due to the 61-dimensional vector having features that relate to the information on only one of the axis and thus, study the habits in each direction separately. However, it is worth emphasizing that in both cases the results are close to a random choice. The most likely cause is that the swiping motions are very simple gestures, and if only one feature is considered a lot of users do it the same way. With the SFFS feature set selection algorithm in the 61-dimensional feature set it is observed that most of the features selected are geometry related features, which may be caused by other features, such as speed or time related, not being stable and depending of the purpose of the gesture being made.

The first approach tested here to model users' behaviour and authenticate them has been using a discriminative SVM classifier following the protocol proposed in [2] to obtain a benchmark. The results obtained are at most 10% worse than the ones obtained in that article. However, by implementing three slight variations on the protocol, the performance improves and the EER is better than the one obtained in [2].

The second system uses a statistical classifier, GMM with UBM adaptation. It has been shown that GMM can be used to model user's behaviour. Afterwards, it can be observed that to adapt from the UBM general model to the specific users' data, the GMM components change their weights and move to model the user, covering the areas where new data points from this specific user may fall. The results show that each kind of touch gesture needs a different value for the three parameters that can be adjusted in this system, the relevance parameter r and the number of training examples and, specially, the number of GMM components. The best performing r is, in general, a balance between a high value, where the GMM components barely change with the user's data, and a low value, where they are considerably adapted to the user's feature vectors. The number of training samples needed to adapt the GMM is low, in most cases with just 20 training samples a good performance is obtained.

Even though the performance obtained with GMM-UBM does not improve the one obtained in the discriminative system, it presents some advantages. In the first place, as the distribution

of the data is not taken into consideration, some users cannot be authenticated in the SVM system, because their patterns have a lot of variability, while the GMM system is capable of representing them. Additionally, another advantage is needing less training samples to obtain the user's model, important in a real-case scenario.

The fusion based system proposed combines the information found in both systems. As a result, the performance improves in most cases and a lower EER is found, with figures around 10% EER in most scenarios tested.

To conclude, this work has shown that horizontal swipes appear to be more discriminative than vertical ones, probably due to being shorter and having less variability within the same user. It can be observed that the overall performance of touch operations in landscape orientation is better, probably because the few users who have data in this orientation have developed more stable patterns by performing them frequently.

5.2 Future work

Taking into account the results obtained, several lines to follow it up arise:

- Extending the number of gestures used to authenticate to multi-touch gestures, such as pinch or rotation. Previous works have shown that this gestures are discriminative enough to differentiate users [15]. Additionally, hypothesis regarding the geometry of the hand or the fingers used can be applied in this problem.
- Studying the intra-user variability considering different time spans [6] and developing techniques to mitigate it. Take into account a real-life situation in which a user is trained and evaluated in the same day.
- Studying touch data obtained from tablets [10, 16], which have a greater screen and thus greater degrees of freedom. The impact this has in touch biometrics should be explored.
- Studying the performance using specific applications and observing if better performance is obtained compared to the one obtained leaving freedom to do any gesture to the user [6]. Evaluate if the intra-user variability is reduced for a specific application or if a user performs the same gesture, for example, vertical swipes, differently depending on what he is doing, such as reading an email or browsing through the web.
- Analysing possible vulnerabilities of touch biometrics to mimic or other attacks. So far, they have been considered to be invulnerable to shoulder surfing attacks, due to the hidden nature of some of the features. Nevertheless, works in the literature have proven that observing the user's behaviour allows attacker to fool the systems and bypass the security [17]. Furthermore, touch biometrics may be specially vulnerable due to the many overlapping features between users and the high intra-user variability. This means that a general model may be granted access [18].
- Studying a fusion of biometrics to use in combination with swipe motions, such as the typing patterns or the phone movement [3, 9].
- Expanding touch biometrics to touch pad input. This input presents several similarities to input from touchscreens, such as the swiping motions. Then, this work can help study and users behaviour with a touch pad for biometric authentication [19].

References

[1] M. Martinez-Diaz, J. Fierrez, R. P. Krish, and J. Galbally. Mobile signature verification: Feature robustness and performance comparison. *IET Biometrics*, 3(4):267–277, December 2014.

[2] Abdul Serwadda, Vir V. Phoha, and Zibo Wang. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8, 2013.

[3] Hui Xu, Yangfan Zhou, and Michael R. Lyu. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 187–198, 2014.

[4] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, 2013.

[5] Vishal M. Patel, Rama Chellappa, Deepak Chandra, and Brandon Barbello. Continuous user authentication on mobile devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine*, 33(4):49–61, July 2016.

[6] Chao Shen, Yong Zhang, Xiaohong Guan, and Roy A. Maxion. Performance analysis of touch-interaction behavior for active smartphone authentication. *IEEE Transactions on Information Forensics and Security*, 11(3):498 – 513, March 2016.

[7] Upal Mahbub, Sayantan Sarkar, Vishal M. Patel, and Rama Chellappa. Active user authentication for smartphones: A challenge data set and benchmark results. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2016.

[8] Anil K. Jain, Karthik Nandakumar, and Arun Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 79:80 – 105, 2016.

[9] Rajesh Kumar, Vir V. Phoha, and Abdul Serwadda. Continuous authentication of smartphone users by fusing typing swiping and phone movement patterns. In *2016 IEEE 8th International Conference on Biometrics Theory Applications and Systems (BTAS)*, pages 1–8, 2016.

[10] Margit Antal, Zsolt Bokor, and László Zsolt Szabó. Information revealed from scrolling interactions on mobile devices. *Pattern Recognition Letters*, 56:7–13, April 2015.

[11] M. Martinez-Diaz, J. Fierrez, and J. Ortega-Garcia. Universal background models for dynamic signature verification. In *Proc. IEEE Conference on Biometrics: Theory, Applications and Systems, BTAS*, pages 1–6, September 2007.

[12] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. Academic Press, 4th edition, 2008.

[13] Marcos Martinez-Diaz. *Dynamic Signature Verification for Portable Devices.* MSc Thesis, Universidad Autonoma de Madrid, November 2008.

[14] Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12):2270 – 2285, 2005.

[15] Napa Sae-Bae, Nasir Memon, and Katherine Isbister. Investigating multi-touch gestures as a novel biometric modality. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 156–161. IEEE, 2012.

[16] Premkumar Saravanan, Samuel Clarke, Duen Horng (Polo) Chau, and Hongyuan Zha. Latent gesture: Active user authentication through background touch analysis. In *Proceedings of the Second International Symposium of Chinese CHI*, Chinese CHI '14, pages 110–113, New York, NY, USA, 2014.

[17] Hassan Khan, Urs Hengartner, and Daniel Vogel. Targeted mimicry attacks on touch input based implicit authentication schemes. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '16, pages 387–398, New York, NY, USA, 2016.

[18] Abdul Serwadda and Vir V. Phoha. When kids' toys breach mobile phone security. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer Communications Security*, CCS '13, pages 599–610, New York, NY, USA, 2013.

[19] Alexander Chan, Tzipora Halevi, and Nasir Memon. Touchpad input for continuous biometric authentication. In *Proceedings of the 2014 Communications and Multimedia Security*, pages 86–91, Berlin, Heidelberg, 2014.